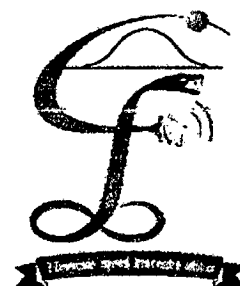
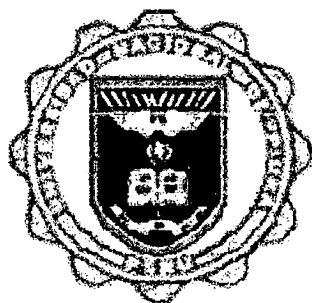


# **UNIVERSIDAD NACIONAL DE PIURA**

**FACULTAD DE CIENCIAS**

**ESCUELA PROFESIONAL DE INGENIERIA ELECTRÓNICA Y  
TELECOMUNICACIONES**



**“DESARROLLO DE UN SISTEMA DE SUPERVISION Y CONTROL DE UNA  
PLANTA APLICANDO COMUNICACIÓN ETHERNET Y TECNOLOGÍA MÓVIL”**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**

**PRESENTADA POR:**

**Bach. ARISTOTELES SONDOR CONCHA**

**Bach. ARNOLD VICENTE JACINTO ECA**

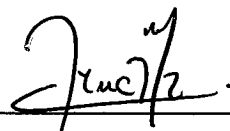
**Piura, Perú  
2015**

**“DESARROLLO DE UN SISTEMA DE SUPERVISION Y CONTROL DE UNA  
PLANTA APLICANDO COMUNICACIÓN ETHERNET Y TECNOLOGÍA MÓVIL”**

TESIS PRESENTADA A LA FACULTAD DE CIENCIAS COMO REQUISITO PARA OPTAR EL TÍTULO  
PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES.



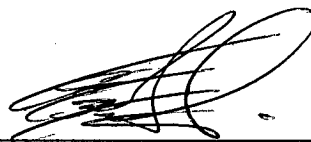
Bach. ARISTOTELES SONDOR CONCHA  
EJECUTOR



Bach. ARNOLD VICENTE JACINTO ECA  
EJECUTOR

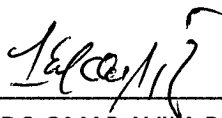


Dr. CARLOS ENRIQUE ARELLANO RAMIREZ  
ASESOR



Ing. HEBERT EDUARDO ESPINO AGUIRRE  
CO-ASESOR

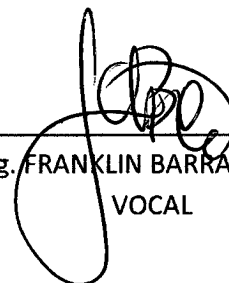
JURADO:



Ing. EDUARDO OMAR AVILA REGALADO  
PRESIDENTE



Ing. JUAN MANUEL JACINTO SANDOVAL  
SECRETARIO



Ing. FRANKLIN BARRA ZAPATA  
VOCAL

Piura, Perú  
2015

### **DEDICATORIA:**

Dedicamos este trabajo principalmente a Dios, por brindarnos la vida y la salud y permitirnos llegar hasta ésta etapa tan importante de nuestra formación profesional.

A nuestros Padres: Albino Jacinto y Sira Eca, Alfredo Sónzor y Cástula Concha, quienes nos trajeron al mundo y nos dieron educación, por ser nuestros guías y por demostrarnos siempre su cariño y apoyo incondicional.

A nuestros familiares, amigos y compañeros de estudios quienes fueron parte importante en nuestra vida universitaria.

### **AGRADECIMIENTOS:**

En primer lugar damos infinitamente gracias a Dios, por habernos dado fuerzas y valor para culminar nuestros estudios superiores de manera satisfactoria y siempre con miras al futuro.

Agradecemos a nuestros padres, hermanos y familiares, que han dado todo el esfuerzo para que podamos dar éste gran paso en nuestras vidas personales y profesionales.

A nuestros docentes universitarios, por la formación académica brindada y por sus sabios consejos durante nuestros estudios y paso por la Universidad.

## INDICE

RESUMEN.....	11
ABSTRACT .....	12
INTRODUCCIÓN.....	13
CAPITULO I. PLANTEAMIENTO DEL PROBLEMA.....	15
1.1 DESCRIPCION DE LA REALIDAD PROBLEMÁTICA.....	15
1.2 DELIMITACION DEL PROBLEMA .....	16
1.3 JUSTIFICACION E IMPORTANCIA DE LA INVESTIGACION.....	17
1.4 ANTECEDENTES DE LA INVESTIGACION .....	17
1.5 LIMITACIONES Y VIABILIDAD DE LA INVESTIGACION.....	22
1.5.1 Limitaciones.....	22
1.5.2 Viabilidad de la investigación.....	22
1.6 FORMULACION DEL PROBLEMA.....	23
1.7 OBJETIVOS DE LA INVESTIGACION.....	23
1.7.1 Objetivo general.....	23
1.7.2 Objetivos específicos.....	23
1.8 HIPOTESIS.....	24
1.9 VARIABLES.....	24
CAPITULO II. MARCO TEORICO.....	25
2.1 SENSORES.....	25
2.1.1 Características de los sensores.....	26
2.1.2 Resolución y precisión.....	27
2.1.3 Tipos de sensores.....	28

2.1.3.1	Sensor de Temperatura LM-35.....	28
2.1.3.2	Sensor de humedad DHT 11.....	30
2.1.3.3	Sensor de ultrasonidos HC-SR04.....	36
2.1.3.4	Sensor de presencia HC-SR501 (PIR).....	39
2.1.3.5	Tecnologías de Autoidentificación y RFID.....	46
2.2	PLATAFORMA DE DESARROLLO: ARDUINO.....	53
2.2.1	Software para Arduino.....	54
2.2.2	Arduino UNO R3.....	57
2.2.3	Arduino MEGA.....	63
2.2.4	Shield Ethernet W5100.....	65
2.3	SISTEMAS DE CONTROL.....	68
2.3.1	Sistemas SCADA.....	69
CAPITULO III. MATERIALES Y METODOS .....		72
3.1	LUGAR DE LA INVESTIGACION.....	72
3.2	DURACION DE LA INVESTIGACION.....	73
3.3	MATERIALES DE LA INVESTIGACION.....	73
3.4	METODOLOGIA DE LA INVESTIGACION.....	75
CAPITULO IV. RESULTADOS .....		76
4.1	DESCRIPCION GENERAL DEL SISTEMA.....	77
4.2	DESARROLLO DEL SOFTWARE DE LAS TARJETAS DE LAS ESTACIONES DEL SISTEMA DE SUPERVISION Y CONTROL.....	79
4.2.1	Módulo de seguridad y proximidad.....	79
4.2.2	Módulo de acceso vía RFID.....	83
4.2.3	Módulo de control de temperatura.....	86
4.2.4	Configuración de la red.....	90
4.2.5	Desarrollo de la aplicación para el dispositivo HMI.....	92

CAPITULO V. ANALISIS Y DISCUSION .....	97
CONCLUSIONES.....	104
RECOMENDACIONES .....	107
REFERENCIAS BIBLIOGRAFICAS .....	109
ANEXOS.....	111
CODIGO EN ARDUINO DEL MODULO DE SEGURIDAD .....	111
CODIGO EN ARDUINO DEL MODULO DE IDENTIFICACION Y	
CONTROL DE ACCESO.....	119
CODIGO EN ARDUINO DEL MODULO DE CONTROL DE	
TEMPERATURA .....	129

## INDICE DE FIGURAS

Figura N°1: Sensor LM35.....	29
Figura N°2: Configuración del sensor de temperatura.....	29
Figura N°3: Sensor DHT11.....	30
Figura N°4: Palabra de control sensor DHT11.....	32
Figura N°5: Diagrama de acoplamiento del sensor DHT11 y Arduino UNO R3.....	34
Figura N°6: Diagrama de flujo de la placa Arduino UNOR3 y el sensor DHT11.....	35
Figura N°7: Vista general del sensor HC-SR04.....	38
Figura N°8: Pines de conexión del sensor HC-SR04.....	38
Figura N°9: Diagrama de conexión sensor HC-SR04 al Arduino.....	39
Figura N°10: Vista general del sensor PIR HC-SR501.....	40
Figura N°11: Diagrama de conexión sensor HC-SR501 al Arduino.....	42
Figura N°12: Pines de conexión del sensor HC-SR501.....	43
Figura N°13: Componentes de un sistema RFID.....	49
Figura N°14: Vista general del módulo RFID-RC522.....	50
Figura N°15: Kit completo del módulo RFID-RC522.....	51
Figura N°16: Diagrama de conexión delRFID-RC522 con Arduino.....	51
Figura N°17: Ejemplo de sistema con RFID.....	53
Figura N°18: Logo oficial de Arduino.....	54
Figura N°19: Interfaz del software Arduino.....	57
Figura N°20: Descripción de la placa de Arduino UNO R3.....	59



Figura N°21: Vista de Arduino MEGA.....	65
Figura N°22: Shield Ethernet W5100.....	67
Figura N°23: Diagrama del sistema de supervisión y control.....	76
Figura N°24: Vista superior de la placa del módulo de seguridad.....	80
Figura N°25: Vista del módulo de seguridad incorporando sirena.....	80
Figura N°26: Módulo de seguridad acoplado al controlador.....	81
Figura N°27: Conexión Ethernet.....	82
Figura N°28: Interfaz del módulo de seguridad.....	82
Figura N°29: Vista superior del módulo RFID.....	83
Figura N°30: Indicadores LED instalados.....	84
Figura N°31: Módulo de RFID acoplado al controlador.....	85
Figura N°32: Conexión Ethernet.....	85
Figura N°33: Interfaz del módulo de acceso vía RFID.....	86
Figura N°34: Diagrama clásico de sistema de control realimentado.....	86
Figura N°35: Vista superior de la placa del módulo de control de temperatura.....	88
Figura N°36: Vista del módulo de control de temperatura completo.....	88
Figura N°37: Conexión Ethernet.....	89
Figura N°38: Vista general del Router P-660HW.....	90
Figura N°39: Vista posterior del Router P-660HW.....	91
Figura N°40: Entorno Web de App Inventor 2.....	92
Figura N°41: Programa en bloques de la aplicación.....	96

## INDICE DE TABLAS

Tabla N°1: Especificaciones técnicas del sensor DHT11.....	31
Tabla N°2: Consideraciones de uso del sensor DHT11.....	33
Tabla N°3: Características técnicas del sensor HC-SR04.....	37
Tabla N°4: Características técnicas del sensor HC-SR501.....	44
Tabla N°5: Características técnicas del Arduino MEGA.....	63

## **RESUMEN**

El desarrollo de la presente tesis está basado en el diseño e implementación de un sistema de supervisión y control de una planta aplicando comunicación Ethernet y tecnología móvil, haciendo uso de una plataforma de desarrollo muy conocida y práctica como lo es Arduino, aprovechamos en hacer uso de las tecnologías libres y sus componentes y diferentes módulos de acoplamiento que permiten desarrollar aplicaciones bastante flexibles y fiables.

Se ha implementado una red LAN con 3 módulos Ethernet, como base para la comunicación y configuración en topología estrella, lo cual se ha realizado con un Switch – Access Point que permite conexiones cableadas mediante conectores RJ – 45 y a la vez conectar dispositivos WI FI que cumplen labor de HMI en el sistema planteado.

El primer módulo está basado en seguridad ocupacional, es decir, se trata de proteger al trabajador ante el acercamiento a equipos o zonas peligrosas en una planta, el segundo módulo está basado en identificación y registro de usuarios y/o trabajadores mediante tecnología RFID y permitir o denegar el acceso según el registro, el tercer módulo es un clásico sistema de control de temperatura mediante sensores. Cabe indicar que la base de la presente investigación es la comunicación en red Ethernet y sobre todo la monitorización en tiempo real mediante interfaces Web que se ejecutan desde cada dispositivo presente en el sistema

## **ABSTRACT**

Development of this thesis is based on the design and implementation of Monitoring and Control System of a plant using Ethernet Communication and Mobile Technology , using a development platform very known and practice as it is Arduino, we take in make use of Free Technologies and its components and different coupling modules that develop sufficiently flexible and reliable applications.

We have implemented a LAN with 3 Ethernet modules, as the basis for communication and star topology configuration, which was performed with a switch - Access Point that can allows wired connections via RJ - 45 and also connect devices WI FI HMI work meet the proposed system .

The first module is based on occupational safety, it is to protect workers against the approach to equipment or dangerous areas in a plant, the second module is based on the identification and registration of users and/or workers with RFID technology and allow or deny access based on the record, the third module is a classic temperature control system using sensors. It is noted that the basis of this research is communication in Ethernet network and especially the real time monitoring via Web interfaces that run from each device in the system.

## INTRODUCCION

Con el avance de la tecnología, el hombre ha logrado inventar y desarrollar sistemas que facilitan el desarrollo de aplicaciones industriales, estos avances tecnológicos, tanto en software como en hardware hacen que los dispositivos electrónicos sean cada vez más sofisticados y puedan estar dotados con inteligencia.

La irrupción de las tecnologías como Ethernet en el plano industrial ha abierto nuevas posibilidades en las aplicaciones de diseño y producción industrial. Así, es clara la tendencia de los últimos años hacia la utilización de sistemas distribuidos de control cada vez más flexibles, fiables y potentes. Esta tendencia es una de las consecuencias de la introducción de las tecnologías de la información, ya que permiten incrementar la fiabilidad y la calidad del producto a la vez que asegurarán una producción eficiente. El uso de los sensores electrónicos permite el desarrollo de plataformas tecnológicas orientadas al sector industrial las cuales facilitan la implementación de sistemas avanzados de control de procesos.

La presente investigación titulada “Desarrollo de un sistema de supervisión y control de una planta aplicando comunicación Ethernet y tecnología móvil” consta de cinco capítulos.

El capítulo I, abarca el planteamiento del problema, el problema, la justificación, objetivos e hipótesis de la investigación, en ese sentido analiza el diseño de un sistema de monitoreo y control que pueda ser aplicado a una planta industrial. Plantea los objetivos a seguir.

El capítulo II, expone el marco teórico, conceptos, procedimientos y técnicas de donde se van a seleccionar las herramientas para el desarrollo del proyecto. Se detallan los modelos existentes de supervisión y control y aplicaciones que usan tecnología Ethernet de las cuales se ha realizado una revisión para proponer un modelo aplicable a una planta industrial.

El capítulo III, establece los materiales y metodología de la investigación que se ha tenido en cuenta para la realización de este proyecto, así mismo se determina el tipo de investigación para el mismo.

El capítulo IV, se desarrolla la propuesta del modelo de sistema de monitoreo y control en tiempo real aprovechando la comunicación Ethernet y la disponibilidad de tecnologías y plataformas de tipo Open Source como lo son los módulos Arduino y sus Shield.

El capítulo V, trata sobre las pruebas realizadas del sistema de monitoreo y control, se tomaron los tiempos de respuesta, se midieron los tamaños de programa para determinar la cantidad de memoria operativa utilizada de los microcontroladores y se determinó la eficacia del modelo propuesto.

Finalmente se realizan las conclusiones de este proyecto y se hacen las recomendaciones respectivas para la optimización futura de la propuesta del modelo expuesto.

## **CAPITULO I. PLANTEAMIENTO DEL PROBLEMA**

### **1.1. DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA**

El actual auge del sector industrial en el Perú ha llevado a la necesidad de implementación de nuevas y cada vez más complejas aplicaciones para soluciones industriales, básicamente el control y monitoreo de sus procesos en forma eficaz y en tiempo real.

Teniendo en cuenta que la adquisición y acceso a soluciones industriales en el mercado es altamente costosa como por ejemplo de los sistemas SCADA, PLD, OPC, etc. Además de poseer arquitecturas cerradas en las cuales no se tiene información de su programación y por ende el soporte técnico de las mismas está condicionado a ser realizado por expertos del equipo del mismo proveedor, lo cual conlleva a depender de quienes realizaron la implementación y configuración.

Debido a ésta problemática, la presente investigación propone el diseño y desarrollo de un modelo de supervisión y control de una planta, que además de resultar de un costo muy bajo y competente en cuanto a resultados y tecnología se refiere, además debe ser considerado que la implementación es totalmente práctica y por ende su mantenimiento y soporte.

## **1.2 DELIMITACION DEL PROBLEMA**

El problema principal radica en que actualmente es cada vez más necesaria la implementación de nuevas tecnologías, sobre todo en el campo industrial y el presente proyecto de investigación apunta a desarrollar un prototipo o piloto que permita monitorear una planta en la cual se puedan obtener variables de Ingeniería de Control, como temperatura y humedad, además de dispositivos de Seguridad Industrial como sensores de CO<sub>2</sub> y de proximidad (PIR), los cuales van a estar integrados en una red de sensores que utilizarán como Sistema de Adquisición una Placa de Desarrollo Arduino UNO R3, y finalmente el monitoreo será realizado mediante una Shield Ethernet W5100 para plataforma de Arduino, la cual trabaja bajo los parámetros del Protocolo TCP-IP y permita la comunicación en red integrado mediante una Topología Estrella.

En cuanto a delimitación, el presente proyecto está orientado a ser implementado para cualquier tipo de planta que pueda ser sometida a parámetros de control y de aplicaciones industriales, pero para efectos de demostración y sustentación a la finalización de la investigación, estará delimitado a una maqueta a escala de una planta de control industrial.



### **1.3 JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN**

La importancia y justificación de la presente investigación radica en realizar el estudio para la obtención de un modelo base de Sistema de Monitoreo y Control en tiempo real de una planta que fácilmente puede ser implementada a nivel industrial utilizando tecnologías de Open Source, tanto en Hardware como en Software Libre, ya que la aplicación puede ser dimensionada en función a la necesidad de una solución.

Además, analizando desde el punto de vista profesional, se desarrolla el presente trabajo porque genera gran interés en el investigador, debido a la factibilidad para su diseño e implementación, por ser un tema de actualidad tecnológica que permite el desarrollo de las capacidades y destrezas investigativas y genera una proyección al futuro.

### **1.4. ANTECEDENTES DE LA INVESTIGACIÓN**

**Contreras, Anguie (2010)**, en su tesis “Diseño de un sistema de automatización para el sistema de iluminación de una planta industrial” nos dice: El sector industrial se constituye hoy en día en un potencial consumidor de energía eléctrica, sin embargo muchas de sus instalaciones no cuentan con un control y uso adecuado de energía. Muestra de ello son los sistemas de iluminación, pues se han identificado numerosas deficiencias en torno a su uso siendo la más crítica mantener encendidas las lámparas durante periodos no

productivos, lo cual conlleva a incrementar el impacto económico para la empresa y ambiental para el mundo.

En la presente tesis se diseña un sistema de automatización para el sistema de iluminación de una planta industrial que permita optimizar el uso de energía eléctrica evitando que las lámparas permanezcan encendidas durante períodos no productivos. Es decir, se diseñará la automatización de alumbrado que asegure una iluminación de calidad durante el tiempo que sea necesario. El diseño consta de la selección de sensores y actuadores, diseño de un programa en lenguaje “Ladder” que permita controlar la secuencia de encendido de los contactores de fuerza de los circuitos de iluminación en función a los períodos de producción y la señal de sensores en las zonas elegidas. Así mismo se ha diseñado la interfaz de usuario la cual permite: ingresar los horarios de producción, realizar el control manual de los circuitos de iluminación desde la interfaz y monitoreo de horas de consumo. El usuario podrá controlar el sistema de iluminación a través de la interfaz de usuario la cual estará comunicada con el controlador a través de una red industrial que también será diseñada. Como parte del desarrollo de la tesis, se hicieron pruebas aisladas y se comprobó el correcto funcionamiento del programa implementado mediante la comunicación de una computadora personal con el controlador, así mismo, las pruebas realizadas verifican la respuesta en tiempo real del controlador empleado en el diseño.

**Nelson Guamán (2012)**, presenta en la Escuela Politécnica Nacional de Quito –Ecuador, su proyecto de investigación denominado “Diseño e implementación de un sistema SCADA en las plantas de tratamiento y compresión de gas natural”.

El presente proyecto tiene por objetivo diseñar e implementar un sistema SCADA que permita supervisar el funcionamiento de las plantas compresoras de gas natural tigre y 67 del bloque Gustavo Galindo Velasco desde una estación central y dos estaciones remotas.

En la estación central se hace el control supervisor hacia las plantas y desde cada estación remota se hace la adquisición de datos desde las plantas. El sistema de control es realizado por un PLC Twido ubicado en cada estación, el cual se programa para leer los registros de memoria de la variable del proceso de las plantas y se le configura para conectarlo a la red LAN de la empresa.

La interfaz hombre – Máquina (HMI) del sistema SCADA se desarrolla en el entorno del software viejo Citect y para establecer la comunicación entre la HMI y el PLC se utiliza Modbus/ TCP. Finalmente, se presenta un sistema SCADA el cual permite un monitoreo remoto y constante de la presión y temperatura de los separadores de la planta, velocidad del motor, parámetro del sistema, histórico de alarmas y tendencias de variables, como también permite controlar el aumento o disminución de los rpm del motor y un paro de emergencia.

Se tomó el presente proyecto de investigación como antecedente porque la automatización en el cual permite y tener una supervisión, control de las Plantas motores – compresores y monitoreo de la presión y temperatura de los separadores el cual es un aporte muy importante para la empresa y para la seguridad de la planta y sus operarios.

**María Hernández y Denis Ledesma (2010),** presenta en la Escuela Superior Politécnica de Chimborazo – Ecuador, su proyecto de investigación denominado “Desarrollo de un sistema de automatización (SCADA) para medición de voltajes con sistemas embebidos para el laboratorio de Mecatrónica de la Facultad de Mecánica”.

El objetivo desarrollado es un sistema SCADA (Supervisión Control y Adquisición De Datos) para la medición de voltajes con sistemas embebidos para el Laboratorio de Mecatrónica de la Facultad de Mecánica. Con la finalidad de supervisar, controlar y adquirir datos de voltaje se elaboró una guía de prácticas con ejemplos de aplicaciones, procediendo al análisis de cada uno de los manuales para conocer el principio de funcionamiento, montaje, cableado y programación del sistema.

Este sistema consta del hardware embebido de control y adquisición de datos NIcRIO -9074 que posee ocho módulos conectables de entrada y salida con su respectivo software el cual se utilizó para establecer la comunicación con la computadora. La supervisión y control se realizó

mediante el software Lab VIEW que permitió elaborar el interfaz hombre-máquina (HMI) para operar los elementos de campo.

Como resultado de los ejemplos de aplicación, basados en control PWM, control PID y control FUZZY LOGIC; se pudo entender el principio de funcionamiento, la configuración, la programación y operación del sistema. Se pudo comprobar los beneficios del hardware Compact RIO, se adquirió mayores conocimientos del software Lab VIEW e implementación de un equipo de alta tecnología para prácticas estudiantiles en el laboratorio de Mecatrónica.

Se puede concluir que este sistema permite monitorear, controlar y registrar de una manera rápida, en tiempo real e histórico el parámetro principal de los ejemplos de la aplicación, con lo que se pueda autorizar procesos aplicando nuevas tecnologías.

## **1.5. LIMITACIONES Y VIABILIDAD DE LA INVESTIGACIÓN**

### **1.5.1. Limitaciones**

Las posibles limitaciones que se pueden dar:

- Una de las limitaciones más significativas es la poca información de Sistemas de Monitoreo y Control utilizando funciones de red Ethernet, que se dispone, además de la escasa investigación en éste campo de las Comunicaciones industriales.
- Para efectos de implementación real, una limitación es la poca confianza de las empresas en aplicaciones que no están respaldadas por marcas grandes de Sistemas Industriales.

### **1.5.2 Viabilidad de la investigación**

Luego de haber evaluado la propuesta, se concluye que la realización del proyecto es totalmente viable ya que tanto la barrera económica para la adquisición de dispositivos y la barrera tecnológica han sido superadas por ser una aplicación de bajo costo además que los dispositivos a programar presentan un nivel alto en cuanto a interfaces cliente-servidor.

## **1.6. FORMULACIÓN DEL PROBLEMA**

¿Será posible desarrollar un sistema de supervisión y control de una planta usando Comunicación Ethernet y Tecnología Móvil?

## **1.7. OBJETIVOS DE LA INVESTIGACIÓN**

### **1.7.1. Objetivo general**

Desarrollar un Sistema de Supervisión y Control de una Planta usando Comunicación Ethernet y Tecnología Móvil.

### **1.7.2. Objetivos específicos**

- Implementar el sistema de adquisición de datos mediante sensores que se encargarán de la recolección de datos y parámetros ambientales, que serán objeto de acondicionamiento para su envío en tiempo real, hacia el servidor.
- Desarrollo del software de los controladores y tarjetas de adquisición, los cuales realizan el enlace y sincronización desde los puertos del hardware hasta el servidor web.

- Definir la arquitectura de la Red y su implementación basada en el modelo OSI y protocolo de comunicación TCP-IP, para la solución al problema de comunicación.
- Diseñar las interfaces de usuario final basado en tecnologías Web para los equipos de cómputo y sistema Operativo Android para los dispositivos móviles de la red.

### **1.8. HIPÓTESIS**

Es posible y viable llevar a cabo el desarrollo de un Sistema de Supervisión y Control de una Planta usando Comunicación Ethernet y Tecnología Móvil.

### **1.9. VARIABLES**

- Variable Independiente:

Comunicación Ethernet y Tecnología Móvil

- Variable Dependiente:

Sistema de Supervisión y Control



## **CAPITULO II. MARCO TEORICO**

### **2.1 SENSORES**

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad luminica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación con lo que puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como por ejemplo el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra. Áreas de aplicación de los sensores: Industria automotriz, Industria aeroespacial, Medicina, Industria de manufactura, Robótica, etc.

Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a una base de datos, la toma de valores desde el sensor, etc.

### 2.1.1 Características de los sensores

Entre las características técnicas de un sensor destacan las siguientes:

- Rango de medida: Dominio en la magnitud medida en el que puede aplicarse el sensor.
- Precisión: Es el error de medida máximo esperado.
- Offset o desviación de cero: Valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el offset.
- Linealidad o correlación lineal.
- Sensibilidad de un sensor: Relación entre la variación de la magnitud de salida y la variación de la magnitud de entrada.
- Resolución: Mínima variación de la magnitud de entrada que puede apreciarse a la salida.
- Rapidez de respuesta: Puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- Derivas: Son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- Repetitividad: Error esperado al repetir varias veces la misma medida.

Un sensor es un tipo de transductor que transforma la magnitud que se quiere medir o controlar, en otra, que facilita su medida. Pueden ser de indicación directa (ejemplo: un termómetro de mercurio) o pueden estar conectados a un indicador (posiblemente a través de un convertidor analógico a digital, un computador y un display) de modo que los valores detectados puedan ser leídos por un humano.

Por lo general, la señal de salida de estos sensores no es apta para su lectura directa y a veces tampoco para su procesamiento, por lo que se usa un circuito de acondicionamiento, como por ejemplo un puente de Wheatstone, amplificadores y filtros electrónicos que adaptan la señal a los niveles apropiados para el resto de la circuitería.

### **2.1.2 Resolución y precisión**

La resolución de un sensor es el menor cambio en la magnitud de entrada que se aprecia en la magnitud de salida. Sin embargo, la precisión es el máximo error esperado en la medida.

La resolución puede ser de menor valor que la precisión. Por ejemplo, si al medir una distancia la resolución es de 0,01 mm, pero la precisión es de 1 mm, entonces pueden apreciarse variaciones en la distancia medida de 0,01 mm, pero no puede asegurarse que haya un error de medición menor a 1 mm. En la mayoría de los casos este exceso de resolución conlleva a un exceso innecesario en el coste del sistema. No obstante, en estos sistemas, si el error en la medida sigue una distribución normal o similar, lo cual es frecuente en errores accidentales, es decir, no sistemáticos, la repetitividad podría ser de un valor inferior a la precisión.

Sin embargo, la precisión no puede ser de un valor inferior a la resolución, pues no puede asegurarse que el error en la medida sea menor a la mínima variación en la magnitud de entrada que puede observarse en la magnitud de salida.

### **2.1.3. Tipos de sensores:**

#### **2.1.3.1. Sensor de temperatura LM-35 (Flores, 2013)**

LM35 es un sensor de temperatura integrado de precisión, cuya tensión de salida es linealmente proporcional a temperatura en °C (grados centígrados). El LM35 por lo tanto tiene una ventaja sobre los sensores de temperatura linealmente calibrada en grados Kelvin: que el usuario no está obligado a restar una ganancia constante para obtener grados centígrados. El LM35 no requiere ninguna calibración externa o ajuste para proporcionar una precisión típica de  $\pm 1.4$  °C a temperatura ambiente y  $\pm 3.4$  °C a lo largo de su rango de temperatura (de -55 a 150 °C).

#### **CARACTERÍSTICAS**

- Calibrado directamente en grados Celsius (Centígrados)
- Factor de escala lineal de  $+10 \text{ mV} / ^\circ\text{C}$
- $0.5^\circ\text{C}$  de precisión a  $+25^\circ\text{C}$
- Rango de trabajo:  $-55^\circ\text{C}$  a  $+150^\circ\text{C}$
- Adecuado para aplicaciones remotas
- Bajo coste
- Funciona con alimentaciones entre 4V y 30V
- Menos de  $60 \mu\text{A}$  de consumo
- Bajo auto-calentamiento ( $0.08^\circ\text{C}$  en aire estático)
- Baja impedancia de salida,  $0.1\text{W}$  para cargas de  $1\text{mA}$

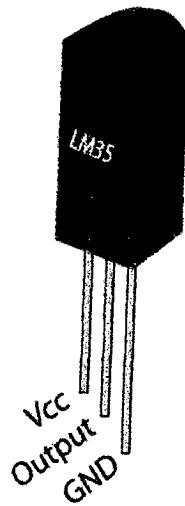


Figura N°1: Sensor LM35

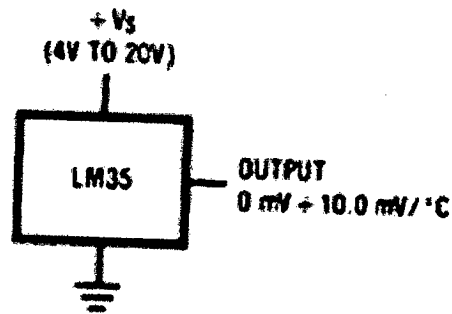


Figura N°2: Configuración del sensor de temperatura

Como se puede ver en la Fig. N°2, esta configuración está calibrada para que el sensor de temperatura mida rangos de temperatura de +2°C a 150°C por lo que son temperaturas que están dentro del rango de operación del proyecto, pues decíamos que el sistema está diseñado para controlar temperaturas por encima de la temperatura ambiente de un clima tropical (para temperaturas negativas el fabricante propone otro tipo de configuración de conexión del sensor de temperatura).

### 2.1.3.2. Sensor de humedad DHT-11 (Flores, 2013)

El DHT11 es un sensor de temperatura y humedad relativa (RH) que integra un sensor con una señal de salida digital calibrada que garantiza una fiabilidad y una estabilidad a largo plazo. El sensor incluye un componente de medición de tipo resistivo con un coeficiente de temperatura negativo que ofrece respuesta rápida, del orden de 6 30 seg (ver Tabla N°1), y una efectividad a bajo costo. En la figura N° 3 se muestra la imagen del sensor.

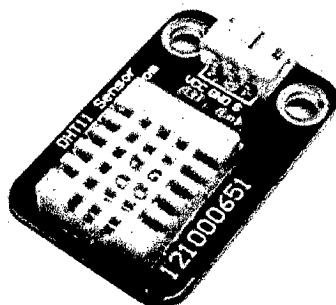


Figura N°3: SensorDHT11

El sensor DHT11 es calibrado por el fabricante en laboratorios para la calibración de humedad. Los coeficientes de calibración son almacenados como datos en la memoria OTP14, que son utilizados por el procedimiento de detección de la variable detectada por el sensor. El cable de interfaz serial hace la conectividad con la placa Arduino UNO R3. El encapsulado cuenta con 4 pines, un bajo consumo de energía que oscila en el rango de 100  $\mu$ A a 2.5 mA y una transmisión de señal de hasta 20 m, por su versatilidad y bajo consumo de energía se emplea en este móvil electrónico.

El formato de datos de un solo bus que es usado para la comunicación y sincronización entre el DHT11 y la placa Arduino Uno R3. Cada proceso de comunicación dura aproximadamente 4ms.

Parámetros	Condiciones	Mínimo	Típico	Máximo
<b>CARACTERÍSTICAS ELÉCTRICAS</b>				
<b>Fuente de alimentación</b>	DC	3 V	5 V	5.5 V
<b>Corriente de alimentación</b>	Funcionando	0.5 mA		2.5 mA
	Promedio	0.2 mA		1 mA
	Espera	100 $\mu$ A		150 $\mu$ A
<b>HUMEDAD</b>				
<b>Resolución</b>		1% RH	1% RH	1% RH
<b>Precisión</b>	25°C		$\pm 4\%$ RH	
	20 – 90% RH			$\pm 5\%$ RH
<b>Tiempo de respuesta (segundos)</b>	1/e (63%) 25°C, 1m/s Aire	6 s	10 s	15 s
<b>Estabilidad de largo plazo</b>	Típicas		$\pm 1$ RH/año	
<b>TEMPERATURA</b>				
<b>Resolución</b>		1° C	1°C	1°C
<b>Precisión</b>	0 - 50°C	$\pm 1^\circ\text{C}$		$\pm 2^\circ\text{C}$
<b>Tiempo de respuesta (segundos)</b>	1/e (63%)	6s		30s

Tabla N°1: Especificaciones técnicas del sensor DHT11

La información consiste en partes enteras y decimales. Con una completa transmisión de información de 40 bits, el sensor envía primero el bit más alto.

**Formato de datos:** 8 bits de información entera RH + 8 bits de información decimal RH + 8 bits de información entera T + 8 bits de información decimal T + 8 bits de comprobación de suma. Si la transmisión de datos es correcta, la comprobación de suma deberá ser los últimos 8 bits de “8 bits de información entera RH + 8 bits de información decimal RH + 8 bits de información entera T + 8 bits de información decimal T”. En la figura 4 se muestra la palabra de control.

8 bits enteros RH	8 bits decimales RH	8 bits Enteros T	8 bits Decimales T	8 bits comprobación
----------------------	------------------------	---------------------	-----------------------	------------------------

Figura N°4: Palabra de control sensor DHT11

En la tabla N°2. Se describen algunas consideraciones de uso del sensor.



<b>CONSIDERACIONES TÉCNICAS</b>
El DHT11 utiliza una comunicación 1-wire <sup>15</sup> con la placa Arduino UNO R3.
El vapor de materiales químicos puede producir interferencia con los elementos sensitivos del DHT11 y disminuye su sensibilidad. El alto grado de contaminación química puede producir un daño permanente.
El DHT11 debe de instalarse en el lugar más lejano de las partes de la placa Arduino que generen más calor.
La exposición de tiempo prolongado a la luz solar y ultravioleta puede degradar el rendimiento del DHT11.
La calidad de los cables de conexión afectará la calidad y distancia de comunicación. Es recomendable el uso de pistas.

Tabla N°2: Consideraciones de uso del sensor DHT11.

Para lograr el objetivo de monitorear en tiempo real de la temperatura y humedad, en la figura 5 se presenta el diagrama a bloques propuesto de acoplamiento entre el sensor DHT 11 y la placa Arduino UNO R3 a través del pin digital número 9 (D9) con una comunicación 1-wire.

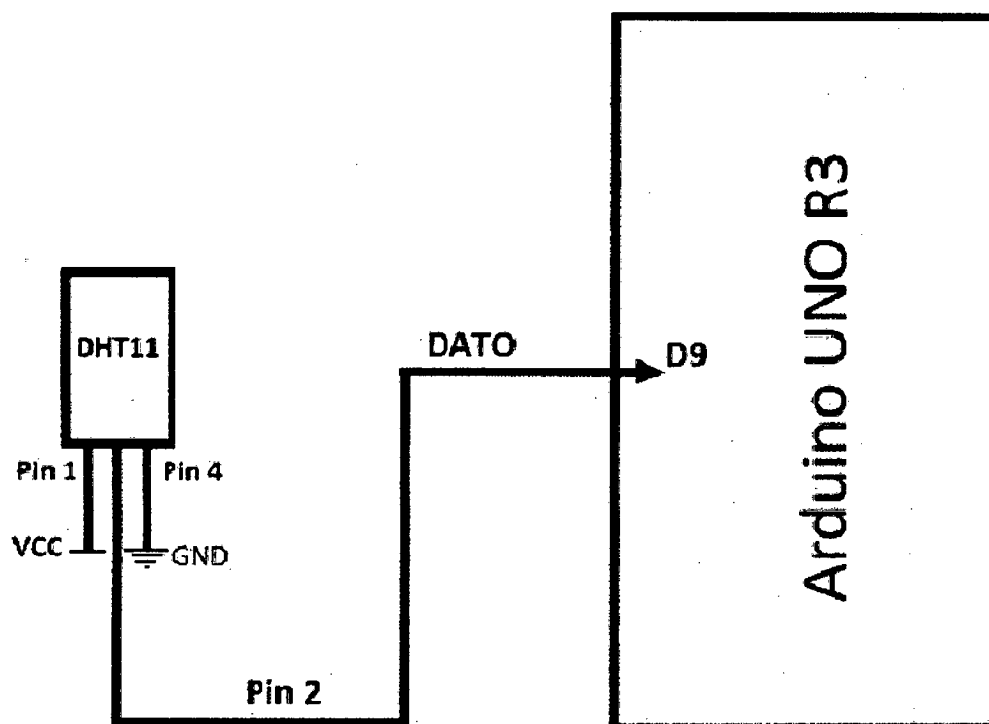


Figura N°5: Diagrama de acoplamiento del sensor DHT11 y Arduino UNO R3

En la figura N°6, se muestra el diagrama de flujo para obtener las variables de temperatura y humedad. Se ajusta la placa Arduino UNO R3 para la recepción de datos a través del puerto digital 9 (D9), activando a su vez el sensor DHT11. Al detectar la señal digital, comienza a sensar las variables (temperatura y humedad) y manda dichos datos obtenidos por su pin digital a la placa Arduino UNO R3. Si no es encontrada la señal no sensani transmite datos. El programa se ejecuta de manera cíclica.

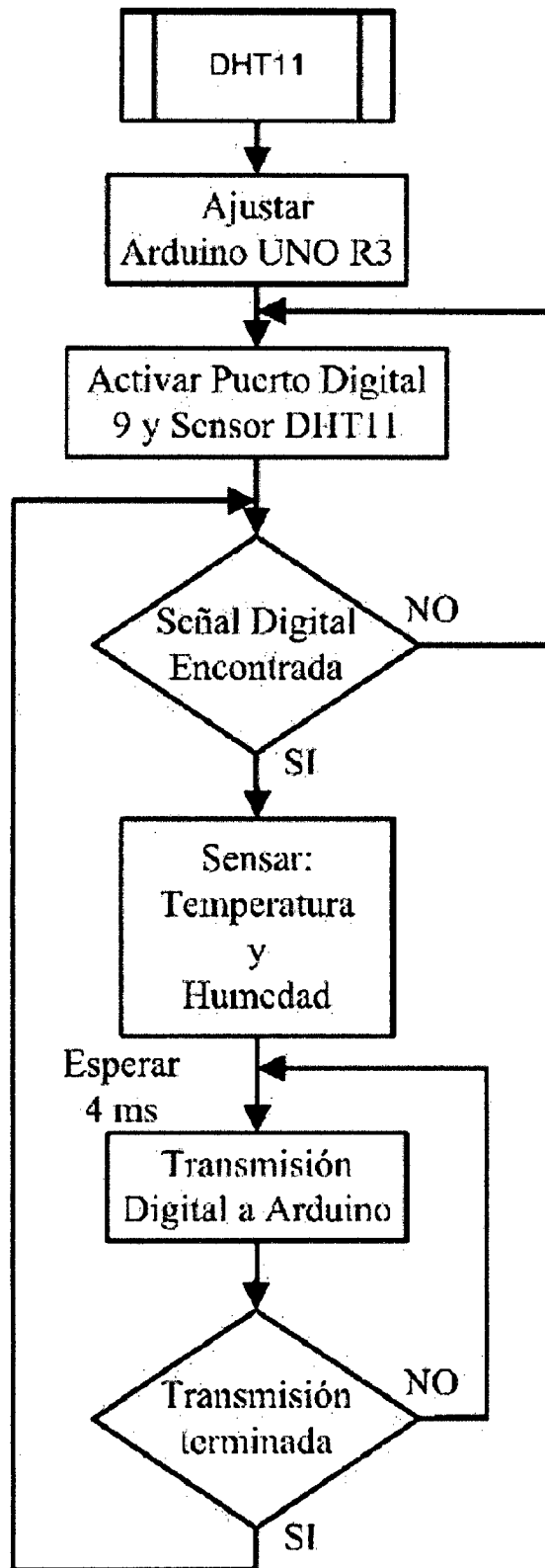


Figura N°6: Diagrama de flujo de la placa Arduino UNO R3 y el sensor DHT 11

### **2.1.3.3. Sensor de ultrasonidos HC-SR04. (Xicohténcal, 2010)**

Los ultrasonidos son señales acústicas cuyas frecuencias están por encima del rango de frecuencias sensibles al oído humano. Los sensores de ultrasonidos son capaces de medir la distancia a la que están respecto a un objeto por medio de un sistema de medición de ecos.

Estos sensores están formados por un transductor que emite un pulso corto de energía ultrasónica, cuando el pulso es reflejado por un objeto, el sensor captura el eco producido por medio de un receptor, y mediante un sistema de tratamiento de la señal, calcula la distancia a la que está de dicho objeto.

Debido a sus características, se pueden encontrar sensores de ultrasonidos en aplicaciones como medición de nivel (en tanques que contienen diferentes productos en forma líquida), control de colisiones en sistemas de aparcamiento, control de posición en campos como robótica, industria del plástico, control de llenado de tanques. Las principales ventajas de estos sensores se basan en que no necesitan contacto físico para poder detectar objetos, tienen una buena relación de calidad-precio y en comparación con otras tecnologías, los dispositivos basados en ultrasonidos son compactos y livianos.

El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de

retorno. De muy pequeño tamaño, destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes, de fácil uso y programación con las placas de **Arduino** y **microcontroladores**.

El HC-SR04 está formado por un emisor y un receptor de ultrasonidos que opera a una frecuencia de 40kHz. El sensor se alimenta a 5V, por lo que se puede alimentar directamente desde Arduino, y puede llegar a detectar objetos hasta 5 metros con una resolución de 1cm.

Modelo	HC-SR04
Voltaje de entrada	5V
Corriente estática	<2mA
Frecuencia de operación	20KHz-400KHz
ángulo sensor	<15°
Distancia de detección	2cm~450cm
Precisión	0.3cm
Entrada de señal de disparo	10us impulso TTL
Tamaño	4.5cm x 2.0cm x 1.3cm

Tabla N°3: Características técnicas del sensor HC-SR04

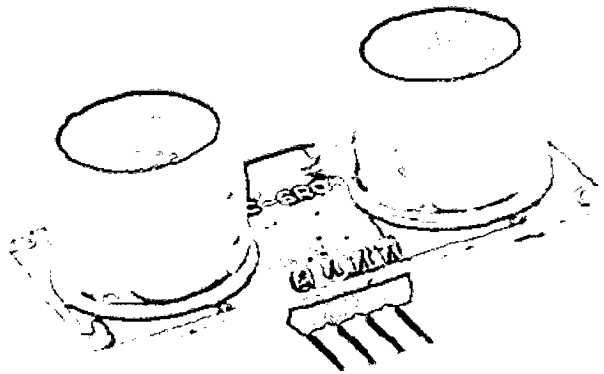


Figura N°7: Vista general del sensor HC-SR04

Pines de conexión:

- VCC
- Trig (*Disparo del ultrasonido*)
- Echo (*Recepción del ultrasonido*)
- GND

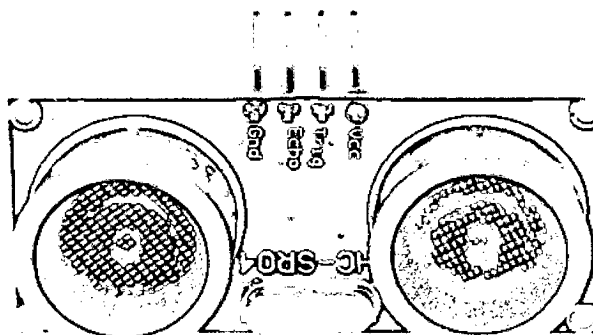


Figura N°8: Pines de conexión del Sensor HC-SR04

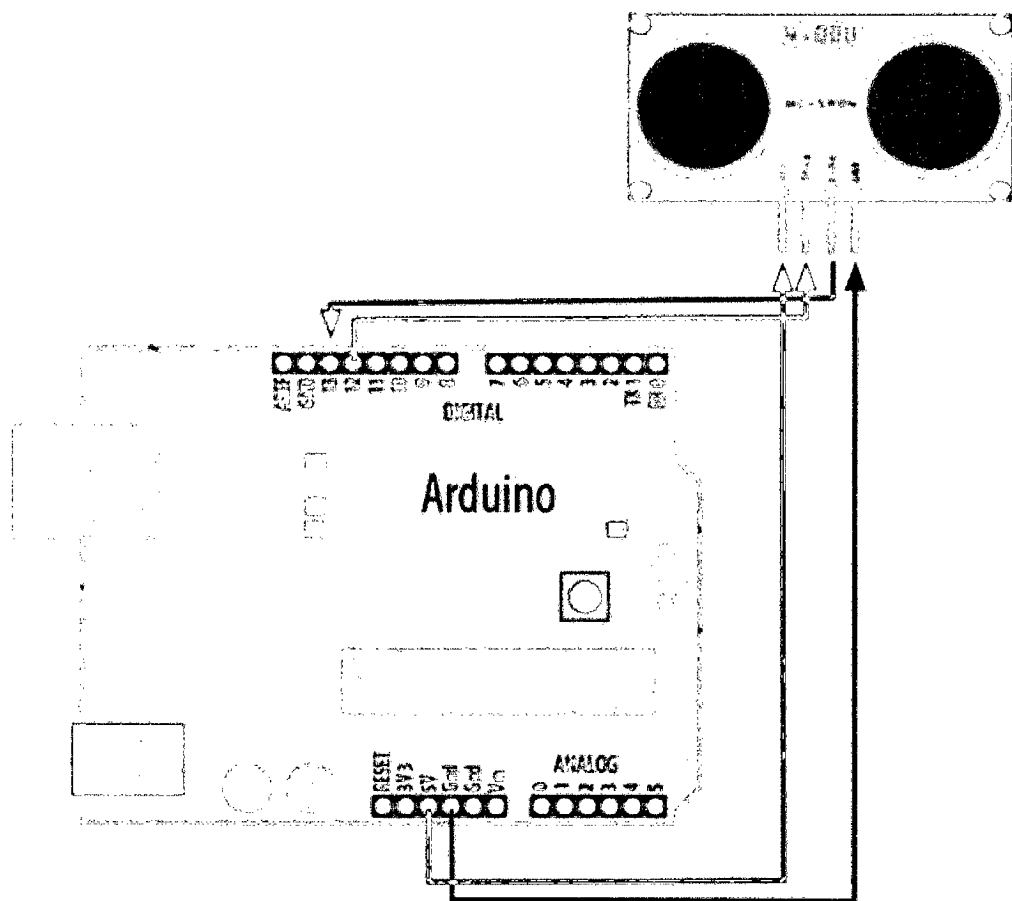


Figura N°9: Diagrama de conexión sensor HC-SR04 al Arduino

#### 2.1.3.4. Sensor de presencia HC-SR501 (PIR).( Marcos, 2013)

A diferencia de otros sistemas, como los ultrasónicos o de microondas, estos sensores no emiten radiación sino que solo reciben "pasivamente" la radiación infrarroja proveniente de cuerpos a temperatura superior al ambiente (todo cuerpo caliente emite radiación infrarroja). Este principio se aprovecha para detectar la presencia de personas sensando la radiación emitida por las mismas. Para tal fin se desarrollaron los detectores piroeléctricos (PIR), que son dispositivos semiconductores capaces de detectar la

radiación infrarroja con tal sensibilidad que una persona puede ser "percibida" a varios metros de distancia. Para que un detector piroeléctrico pueda ser aplicado a un sistema de iluminación automática se debe lograr la captación de variaciones de temperaturas relativas. Una variación térmica de todo el ambiente en su conjunto no indicaría la presencia de una persona como tampoco lo haría la variación de temperatura de un cuerpo. Para evitar disparos no deseados, los detectores en su interior poseen en realidad dos elementos sensibles conectados en oposición de manera que su potencial se cancela cuando la variación térmica los afecta a ambos por igual.

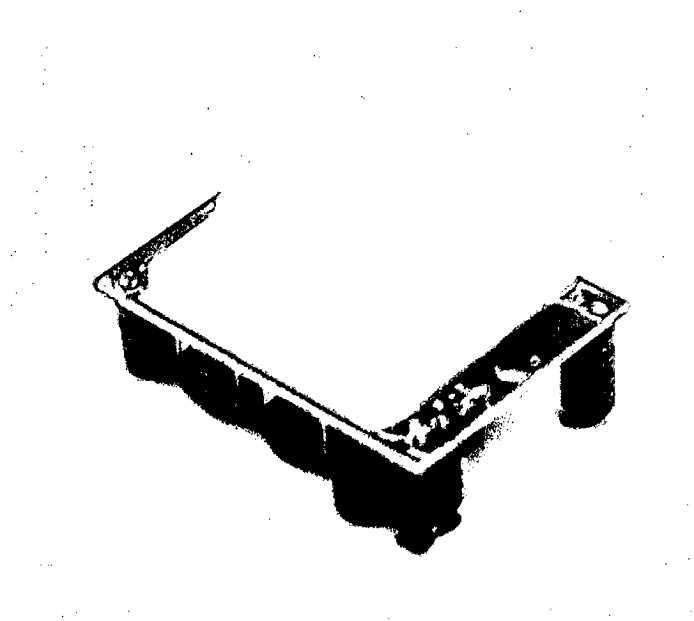


Figura N°10: Vista general del sensor PIR HC-SR501



## VENTAJAS

- **Gran ahorro de energía:** No permite que las luces queden encendidas como sucede típicamente en lugares de ocupación ocasional (baños, aulas, salas, oficinas, etc.) e impide que estas se enciendan durante el día.
- **Disuasión:** El sensor se activa con independencia de que la persona detectada sea o no un intruso. De esta manera el encendido inesperado de las luces sorprenderá al intruso sin saber si fue visto o no, al menos facilitará su observación (lugares de acceso como pasillos, fondos, etc.).
- **Confort y seguridad:** El sistema ofrece la comodidad de encender la luz sin intervención del usuario, evitando muchas veces andar tanteando la pared en la oscuridad hasta dar con la llave de luz, **evitando el riesgo de accidentes por golpes o tropezones por caminar a oscuras.**
- **Protección del medio ambiente:** Es una verdad conocida por todos que la generación de energía es contaminante. Todos los sistemas de generación de energía dañan de una manera u otra nuestro medio ambiente y consumen recursos no renovables, por lo tanto, todos los sistemas que conduzcan a economizar energía, más allá del ahorro del servicio, proporcionan un beneficio colectivo cada vez más necesario.

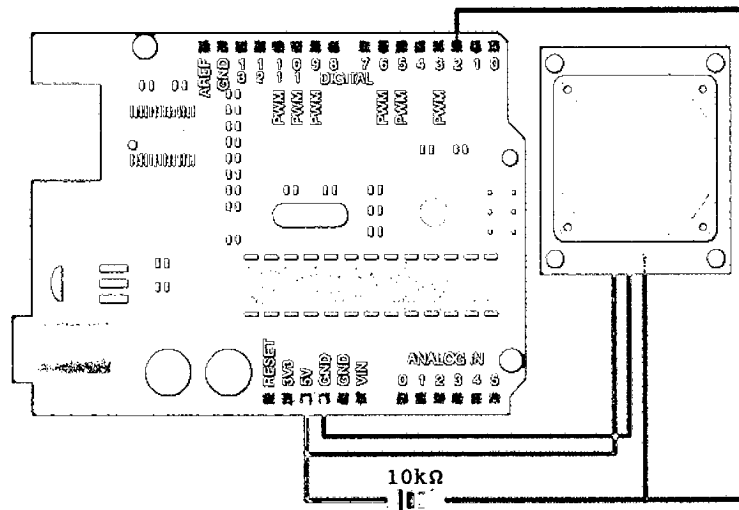


Figura N°11: Diagrama de conexión sensor HC-SR501 al Arduino

El módulo HC-SR501 tiene 3 pines de conexión +5v, OUT (3,3v) y GND, y dos resistencias variables de calibración (Ch1 y RL2).

Ch1: Con esta resistencia podemos establecer el tiempo que se va a mantener activa la salida del sensor. Una de las principales limitaciones de este módulo es que el tiempo mínimo que se puede establecer es de más o menos 3s. Si cambiamos la resistencia por otra de 100K, podemos bajar el tiempo mínimo a más o menos 0,5 s.

RL2: Esta resistencia variable nos permite establecer la distancia de detección que puede variar entre 3-7m.

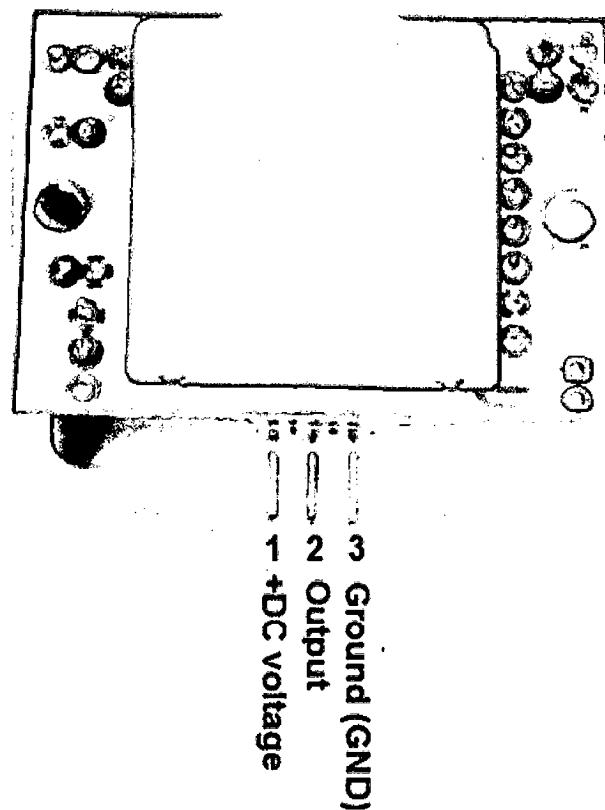


Figura N°12: Pines de conexión del sensor HC-SR501

La posibilidad de mantener activa la salida del módulo durante un tiempo determinado nos permite poder usarlo directamente y prácticamente para cualquier aplicación sin necesidad de usar un microcontrolador.

Tipo de sensor:	completamente pasivo
Rango de detección Alcance:	Aprox. 7m
Ángulo de detección:	< 100 grados
Rango de voltaje de funcionamiento:	4,5V-20VDC
Corriente de consumo en reposo:	menos de 50 $\mu$ A
Voltaje de la señal de salida:	3V Alto / 0V Bajo
Rango de temperatura de funcionamiento:	-15 a +70 °C
Tamaño:	Aprox. 32mm x 24mm

Tabla N°4: Características técnicas del sensor HC-SR501.

### Consideraciones prácticas de instalación

- El vidrio es un filtro infrarrojo, por lo tanto los sensores no detectarán cuando se instalen detrás de puertas de vidrio, ventanas, vidrieras, etc.
- No ubicar en sitios que tengan movimiento o puedan moverse (postes, árboles, etc.).
- No ubicar donde el tránsito es en dirección hacia el sensor, ya que es más sensible en sentido transversal.
- Evitar dirigir el sensor hacia los objetos que puedan ser movidos por el viento

(arbustos, plantas, etc.).

- No es aconsejable dirigir el sensor hacia luces potentes. Si el sensor viene con reflector incorporado, cuidar que la lente del mismo quede por debajo y delante de la lámpara.
- Evitar montar el sensor muy cerca de fuentes de calor (calentadores, secamanos, acondicionadores de aire, etc.).
- Aunque el sensor sea para uso exterior evite que la lluvia directa o nieve incidan sobre la lente.

#### **2.1.3.5. Tecnologías de Autoidentificación y RFID(Alvarado, 2008)**

A lo largo de los años han surgido distintas tecnologías orientadas a la auto identificación. En este capítulo se hace un análisis de las tecnologías de identificación existentes y se presenta un comparativo de sus principales características, a ventajas y desventajas. Se muestra un estudio más profundo de la tecnología de RFID y los componentes necesarios para desarrollar un proyecto con esta tecnología.

##### **a) Tecnologías de Autoidentificación**

Desde hace años están disponibles en el mercado distintas tecnologías para la identificación de productos, personas e incluso animales. En ese sentido, uno de los principales exponentes ha sido el código de barras, el cual ha logrado penetrar

prácticamente en todas las cadenas de distribución, almacenes y sistemas de control de acceso, por citar algunos ejemplos. Sin embargo, en los últimos 10 años, se ha dado un boom de nuevas tecnologías, o más bien de aquellas que ya existían, pero que hasta ahora pudieron entrar al mercado masivo. La razón principal son todas las ventajas tecnológicas que ofrecen frente a los esquemas tradicionales, aunado a la baja en los precios.

#### b) Comparación de Tecnologías de Autoidentificación

Dentro del ámbito de la tecnología de identificación, aplicado al control de acceso, se pueden encontrar diversas tecnologías como: sistemas biométricos, tarjetas magnéticas, código de barras, RFID y memorias de contacto que se describen en los siguientes párrafos.

#### c) Acceso con Tarjetas de RFID (Identificación por Radio Frecuencia)

La tecnología de radiofrecuencia se desarrolló en 1940, como medio para la identificación de los aviones aliados y enemigos durante la Segunda Guerra Mundial. Años más tarde evolucionó, logrando así ser utilizada en la industria ferroviaria para el seguimiento de los coches del ferrocarril y para los años 60's y 70's, su uso se enfocó en la seguridad de materiales nucleares.

En la actualidad RFID se utiliza principalmente en el rubro de seguridad, como es el caso de los cruces fronterizos, credenciales de identidad, en el control vehicular, identificación de ganado, envío de paquetes, control de equipaje en los aeropuertos y de artículos para renta o préstamo (películas y libros) en videoclubes y bibliotecas, en la

industria automotriz, para los procesos de automatización y seguimiento, en el sector agrícola y en el de administración de flora y fauna, para rastrear al ganado y a los animales, así como en el mercado minorista como dispositivo antirrobo. La Tecnología de Identificación por Radiofrecuencia es un método electrónico que consiste en asignar un código de información a un producto, proceso o persona y usar esta información para identificar o acceder a información adicional al respecto.

Los sistemas de identificación por radiofrecuencia consisten generalmente de dos componentes:

- El "transponder", pequeña etiqueta electrónica (tag) que contiene un minúsculo microprocesador y una antena de radio. Esta etiqueta contiene un identificador único que puede ser asociado a una persona o producto.

- El "lector", que obtiene el identificador del "transponder".

La tecnología del transponder se basa en la aplicación de un transmisor/receptor encapsulado. El receptor se puede activar por medio de una batería incorporada (transponder activo) o puede ser alimentado por la señal enviada por el lector (transponder pasivo). El lector genera un campo magnético cuya señal de RF es captada por el receptor del chip. Éste, a su vez activará al transmisor, el cual enviará un mensaje codificado único. Este mensaje es decodificado por el lector y procesado por la computadora.

d) Tecnología RFID

Existen 3 componentes básicos en un sistema de RFID :

1. El tag, etiqueta o transponder de RFID consiste en un pequeño circuito, integrado con una pequeña antena, capaz de transmitir un número de serie único hacia un dispositivo de lectura, como respuesta a una petición. Algunas veces puede incluir una batería.
2. El lector, (el cual puede ser de lectura o lectura/escritura) está compuesto por una antena, un módulo electrónico de radiofrecuencia y un módulo electrónico de control.
3. Un controlador o un equipo anfitrión, comúnmente una PC o Workstation, en la cual corre una base de datos y algún software de control.



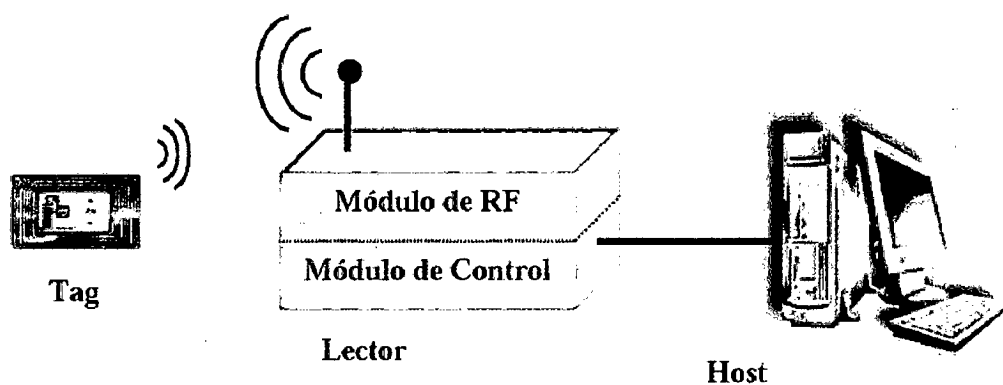


Figura N°13: Componentes de un sistema RFID

e) Módulo Lector RFID-RC522 RF con Interfaz SPI (13.56 Mhz).

El módulo utiliza 3.3V como voltaje de alimentación y se controla a través del protocolo SPI, así como el protocolo UART, por lo que es compatible con casi cualquier micro controlador, Arduino o tarjeta de desarrollo. El RC522 utiliza un sistema avanzado de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz.

Como se hará una lectura y escritura de la tarjeta, es necesario conocer las características de los bloques de memoria una tarjeta: La tarjeta que viene con el módulo RFID cuenta con 64 bloques de memoria (0-63) donde se hace lectura y/o escritura. Cada bloque de memoria tiene la capacidad de almacenar hasta 16 Bytes. El número de serie consiste de 5 valores hexadecimales, se podría utilizar esto para hacer una operación dependiendo del número de serie.

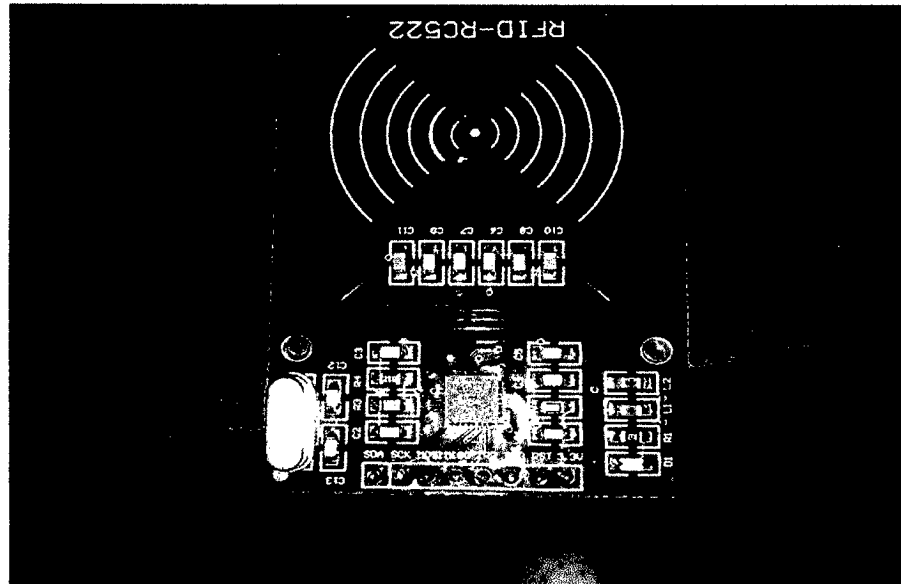


Figura N°14: Vista general del módulo RFID-RC522

#### Características del módulo RFID RC552

- Corriente de operación: 13-26mA a 3.3V
- Corriente de stand by: 10-13mA a 3.3V
- Corriente de sleep-mode: <80uA
- Corriente máxima: 30mA
- Frecuencia de operación: 13.56Mhz
- Distancia de lectura: 0 a 60mm
- Protocolo de comunicación: SPI
- Velocidad de datos máxima: 10Mbit/s
- Dimensiones: 40 x 60 mm
- Temperatura de operación: -20 a 80°C
- Humedad de operación: 5%-95%
- Máxima velocidad de SPI: 10Mbit/s
- Incluye pines, llavero y tarjeta

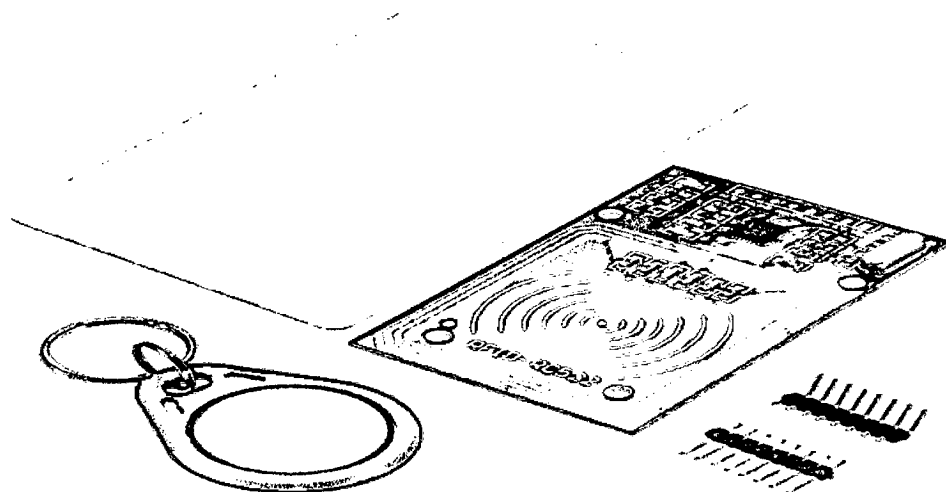


Figura N°15: Kit completo del módulo RFID-RC522

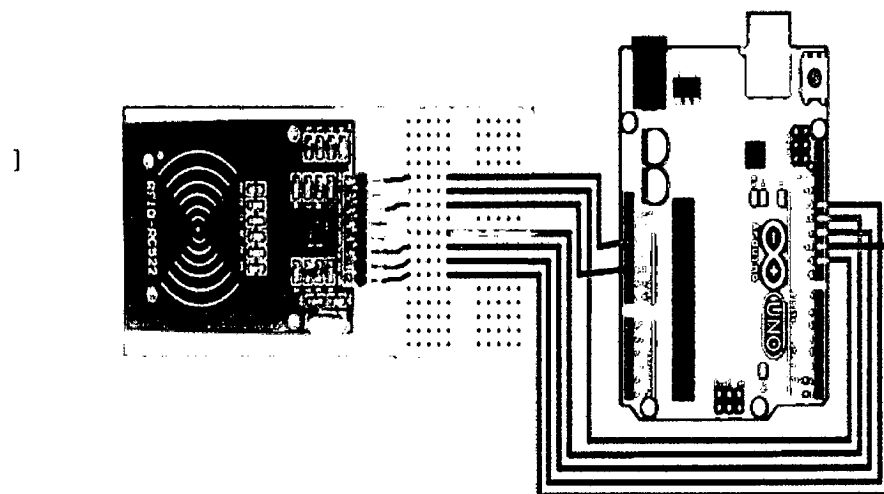


Figura N°16: Diagrama de conexión del RFID-RC522 con Arduino.

Componentes de un sistema RFID:

### **Transmisor**

El transmisor emite potencia y envía el ciclo de reloj a través de su antena hacia los tags que se encuentran dentro de su rango de lectura.

### **Receptor**

Este componente recibe las señales analógicas provenientes del tag a través de la Antena y envía estos datos al microprocesador, donde esta información es convertida en su equivalente digital.

### **Antena**

Esta antena va conectada directamente al transmisor y al receptor. Existen lectores con múltiples puertos para antenas, lo que les permite tener múltiples antenas y extender su cobertura.

### **Microprocesador**

Este componente es responsable de implementar el protocolo de lectura empleado para comunicarse con tags compatibles. Decodifica y realiza verificación de errores a las señales recibidas. Adicionalmente, puede contener cierta lógica para realizar filtrado y procesamiento de bajo nivel de los datos leídos, esto es, eliminar lecturas duplicadas o erróneas.

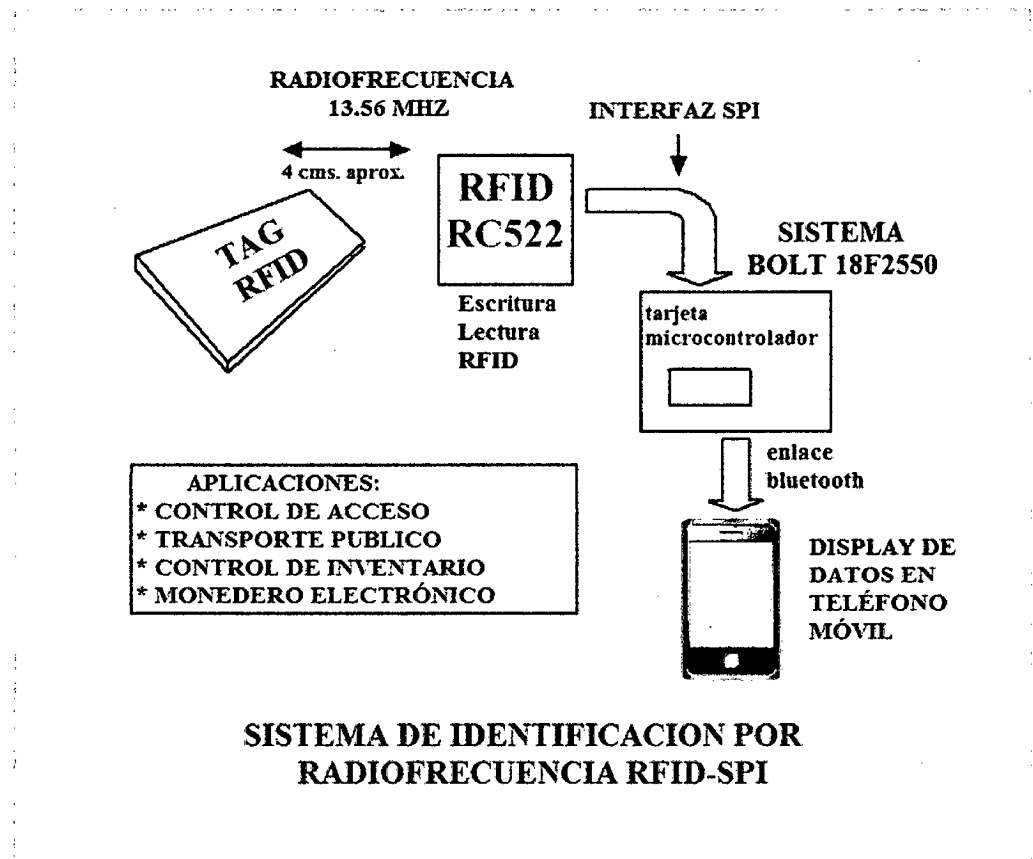


Figura N°17: Ejemplo de sistema con RFID

## 2.2. PLATAFORMA DE DESARROLLO: ARDUINO

Arduino es una plataforma electrónica de código abierto basado en hardware y software fácil de usar. Está dirigido a cualquier persona que hace proyectos interactivos.

Arduino detecta el medio ambiente mediante la recepción de las aportaciones de muchos sensores, y afecta a su entorno por las luces de control, motores y otros actuadores.

Se puede programar Arduino, escribiendo código en el lenguaje de programación de Arduino y utilizando el entorno de desarrollo Arduino.([www.arduino.cc](http://www.arduino.cc)).



Figura N°18: Logo oficial de Arduino  
Fuente: [www.arduino.cc](http://www.arduino.cc)

### **2.2.1. Software para Arduino (Lledó, 2012)**

La plataforma Arduino tiene un lenguaje propio que está basado en C/C++ y por ello soporta las funciones del estándar C y algunas de C++. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino como Java, Processing, Python, Mathematica, Matlab, Perl, Visual Basic, etc. Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario

que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida. Es bastante interesante tener la posibilidad de interactuar con Arduino.

Mediante esta gran variedad de sistemas y lenguaje puesto que dependiendo de cuales sean las necesidades del problema que vamos a resolver podremos aprovecharnos de la gran compatibilidad de comunicación que ofrece.

El entorno de desarrollo de Arduino es sencillo e intuitivo además puede descargarse gratuitamente desde su página oficial para distintos sistemas operativos. Ha sido implementado con Processing, un lenguaje similar a Java. Su última versión es la 1.6.2. Es importante remarcar que la placa Arduino Uno solo la podremos utilizar a partir de la versión beta 0021. Está formado por una serie de menús, una barra de herramientas con botones para las funciones comunes, un editor de texto donde escribiremos el código, un área de mensajes y una consola de texto. En la ilustración 4 se puede apreciar la composición del software de Arduino.

A continuación comentaremos la utilidad de cada área del programa centrándonos solo en lo importante.

### **Menú**

La parte más importante se encuentra en Herramientas. Desde aquí podremos configurar el programa para que pueda comunicarse con la placa Arduino. Pasando el ratón por Tarjeta aparecerá una lista con los tipos de placa Arduino que el programa comprende. Aquí seleccionaremos Arduino Uno o Diecimilia (para la Seeeduino) dependiendo de con cual estemos trabajando. En el campo Puerto Serial

seleccionaremos el que corresponda a nuestra placa que conectaremos mediante USB. Si utilizamos Windows el puerto tendrá un nombre del estilo COMx pero en Linux será /dev/ttyUSBx donde x es un número. En caso de que aparezcan varios puertos serie y no sepamos cual es el de nuestra placa procederemos a desconectarla, anotamos los puertos que aparecen, reconectamos la placa y volvemos a mirar la lista de puertos. El nuevo puerto que haya aparecido será el de nuestra placa.

### **Botones comunes**

Estos botones son accesos rápidos a ciertas acciones que también están disponibles mediante el menú. Los botones son los siguientes:

- Verificar: comprueba y compila el código.
- Cargar: además de compilar el código lo inyecta en la placa.
- Nuevo: crea un nuevo sketch.
- Abrir: abre un sketch previamente guardado.
- Guardar: almacena en disco los cambios realizados en el sketch.
- Monitor Serial: abre una nueva ventana desde la que podremos comunicarnos bidireccionalmente vía serie con la placa, es decir, podremos leer la información que nos envía o proporcionarla nosotros.



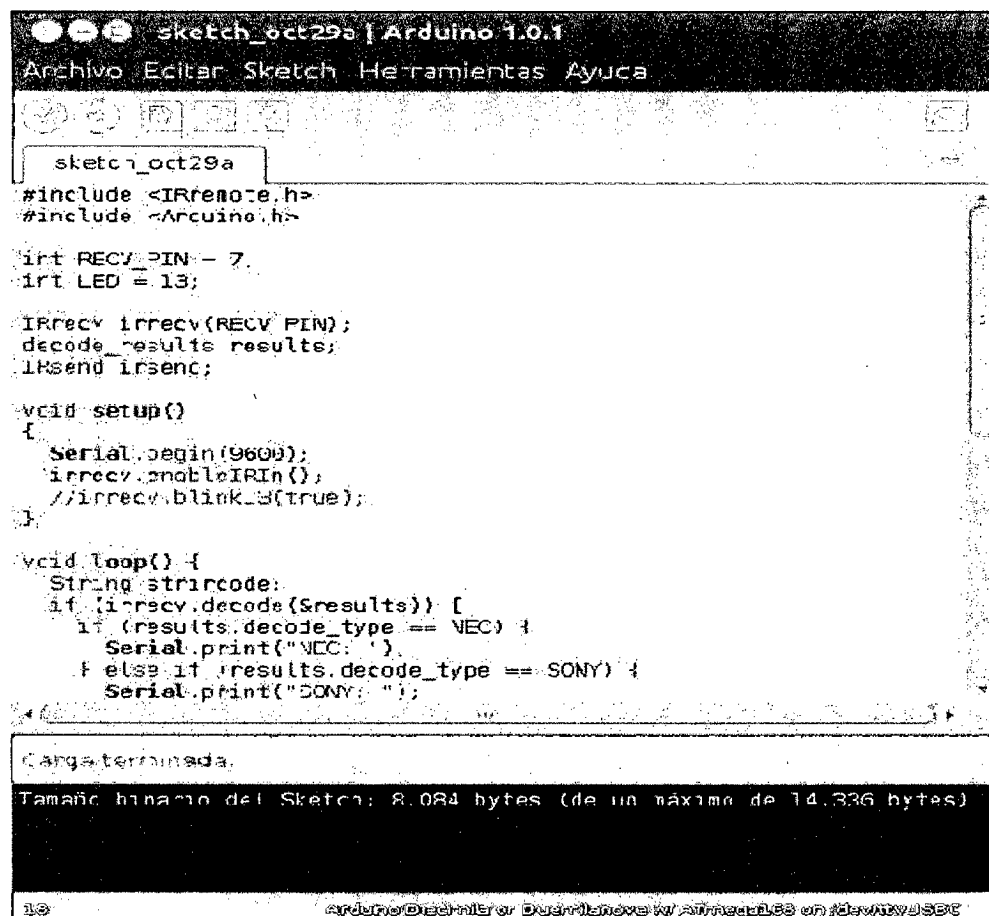


Figura N°19: Interfaz del software Arduino

### 2.2.2. Arduino UNO R3 (Lledó, 2012)

Es el último modelo diseñado y distribuido por la comunidad Arduino. La placa tiene un tamaño de 75x53mm. Su unidad de procesamiento consiste en un microcontrolador ATmega328. Puede ser alimentada mediante USB o alimentación externa y contiene pines tanto analógicos como digitales. La tabla siguiente resume sus componentes:

Microcontrolador:	ATmega328
Voltaje operativo:	5V
Voltaje de entrada (recomendado):	7-12V
Voltaje de entrada (limites):	6-20V
Pines digitales E/S:	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica:	6
Corriente continua para pines E/S	40 mA
Corriente continua para pines de 3.3V	50 mA
Memoria Flash:	32 KB (ATmega328) de los cuales 0.5 KB son para el bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz

A continuación se muestra en la siguiente lustración, donde están ubicados los elementos más importantes que componen la placa Arduino Uno que son descritos de arriba abajo y de izquierda a derecha:

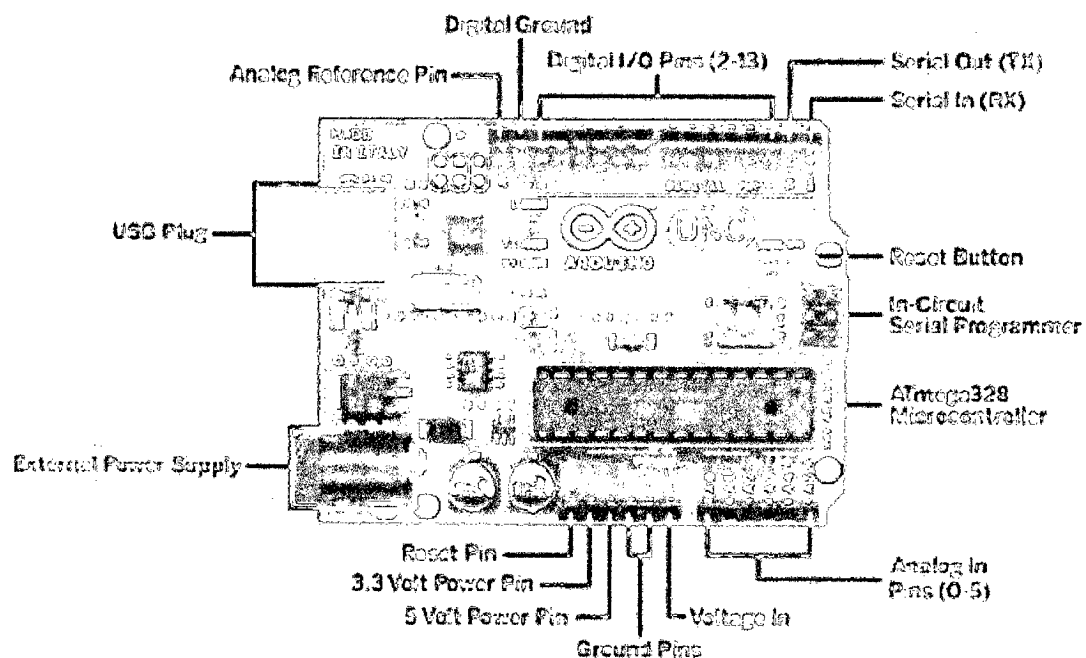


Figura N°20: Descripción de la placa de Arduino UNO R3

### Referencia para pines analógicos (AREF):

Tensión de referencia para entradas analógicas. Se utiliza con la función `analogReference()`.

### Pines de tierra (GND):

Masa del circuito para pines, es decir es la tensión de referencia de 0V.

### Pines digitales de entrada y salida:

En estos pines conectaremos la patilla de dato del sensor/actuador. Desde ellos podremos leer la información del sensor o activar el actuador. Hay 14 pines digitales que pueden utilizarse como entrada o salida con las funciones `pinMode()`,

digitalWrite(), y digitalWrite(). Operan a 5 voltios. Cada pin proporciona o recibe como máximo 40mA y disponen de una resistencia pull-up (desconectada por defecto) de 20-50 kOhmios. Ciertos pines son reservados para determinados usos:

- Serie: 0(RX) y 1(TX). Utilizados para recibir (RX) y transmitir (TX) datos serie. Están directamente conectados a los pines serie del microcontrolador. Utilizando estos pines podremos conectarnos con otras placas.
- Interrupciones externas: 2 y 3. Estos pines pueden ser configurados para activar interrupciones.
- PWM: 3, 5, 6, 9, 10 y 11. Proporcionan una salida de 8 bits en modo PWM.
- SPI: 10-13. Estos pines soportan la librería de comunicación de dispositivos SPI.
- LED: 13. Este pin está conectado con un led de la placa. Cuando se le asigne un valor HIGH se encenderá, en cambio si lo dejamos en LOW estará apagado.

### **Conector USB:**

Existen varios tipos de conectores USB, en concreto esta placa utiliza el tipo B hembra. Con lo cual se necesitará un cable tipo B macho – tipo A macho (aunque se pueden utilizar otros este es el más extendido) que deberá conectarse a un conector tipo A hembra (por ejemplo a un ordenador o al cargador de un móvil). La placa se puede alimentar con la tensión de 5V que le proporciona el bus serie USB. Cuando carguemos

un programa a la placa desde el software de Arduino se inyectará el código del ordenador por este bus.

### **Botón Reset**

Utilizando este botón podremos reiniciar la ejecución del código del microcontrolador.

### **ICSP (In Circuit Serial Programming)**

Es un conector utilizado en los dispositivos PIC para programarlos sin necesidad de tener que retirar el chip del circuito del que forma parte.

### **Microcontrolador ATmega328**

El microcontrolador es el elemento más importante de la placa. Es donde se instalará y ejecutará el código que se haya diseñado. Ha sido creado por la compañía Atmel, tiene un voltaje operativo de 5V, aunque se recomienda como entrada de 7-12V con un límite de 20V. Contiene 14 pines digitales de entrada y salida, 6 pines analógicos que están conectados directamente a los pines de la placa Arduino comentados anteriormente. Dispone de 32KB de memoria flash (de los cuales 512 bytes son utilizados por el bootloader). En la memoria flash se instalará el programa a ejecutar. El bootloader será el encargado de preparar el microcontrolador para que pueda ejecutar nuestro programa. También tiene una memoria EEPROM de 1KB que puede ser leída o escrita con la librería EEPROM. En la parte de procesamiento dispone de un reloj de 16Mhz y 2KB de memoria RAM.

### **Fuente de alimentación externa**

La placa puede ser alimentada también mediante corriente continua suministrada por el conector jack de 3.5mm que podrá recibir entre 7 y 12V.

### **Pin de Reset**

Podemos imitar el funcionamiento del botón reset suministrando un valor LOW(0V) para reiniciar el microcontrolador.

### **Pin de 3.3V**

Desde aquí podremos suministrar 3.3V a los dispositivos que lo necesiten con una corriente máxima de 50mA. Es generada gracias al chip FTDI integrado en la placa.

### **Pin de 5V**

Este pin saca una tensión de 5v del regulador de la placa. El regulador es necesario puesto que puede ser alimentada con distintos voltajes.

### **Pin de Vin**

Es el voltaje de entrada cuando se usa una fuente de alimentación externa (no tiene en cuenta la conexión USB). Se puede proporcionar voltaje a la placa a través de este pin, o en caso de que se esté utilizando una fuente de alimentación externa tomar el valor que está siendo suministrado

### Pines analógicos

Esta placa contiene 6 pines de entrada analógicos. Los elementos que se conecten aquí suelen tener mayor precisión que los digitales pero su uso requiere de una lógica levemente mayor. Más adelante se comentará el uso de un termistor analógico.

### 2.2.3. Arduino MEGA (Lledó, 2012)

El Arduino Mega es una placa electrónica basada en el ATmega1280. Cuenta con 54 pines digitales de entrada / salida (de los cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; simplemente conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC para empezar. La Mega es compatible con la mayoría de los escudos diseñados para el ArduinoDuemilanove o Diecimila.

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
ClockSpeed	16 MHz

Tabla N°5: Características técnicas del Arduino MEGA

El Arduino Mega puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La fuente de potencia externa puede venir con un adaptador de CA a CC, o la batería. El adaptador se puede conectar a un enchufe de 2,1 mm de centro- positivo en el conector de alimentación de la placa. Potenciales de una batería se pueden insertar en los cabezales de pin GND y Vin del conector POWER.

El tablero puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y la junta puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son como sigue:

-Vin: El voltaje de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (por oposición a 5 voltios de la conexión USB u otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de poder, acceder a él a través de este pin.



-5V. La fuente de alimentación regulada utiliza para alimentar el microcontrolador y otros componentes de la placa. Esto puede venir de VIN a través de un regulador de a bordo, o ser suministrada por USB o otra fuente de 5V regulada.

-3V3. Un suministro de 3,3 voltios generado por el chip FTDI de a bordo. Empate Corriente máxima es de 50 mA.

-GND: Pines de tierra.

El microcontrolador ATmega1280 que viene integrado en el Arduino Mega, tiene 128 KB de memoria flash para almacenar código (de los cuales 4 KB se utiliza para el cargador de arranque ), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM) .

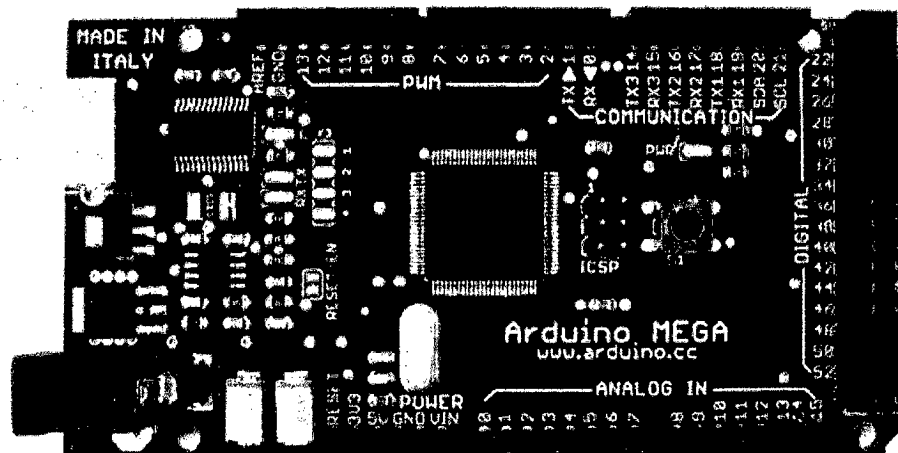


Figura N°21: Vista de Arduino MEGA

#### **2.2.4. Shield Ethernet W5100 (Lledó, 2012)**

La Arduino Ethernet Shield permite que un Arduino se conecte a Internet. Esta Shield está basada en el chip Wiznet W5100, que proporciona un enlace de red (IP) tanto por TCP como por UDP. La Arduino Ethernet Shield soporta la conexión de hasta cuatro elementos de forma simultánea. Puedes usar la librería de Ethernet para escribir programas que se conecten a Internet usando esta Shield.

La Ethernet Shield se conecta a una placa Arduino mediante el largo terminal macho situado en sus laterales. Esto mantiene el esquema de pines intacto y permite enchufar otra Shield encima de ella. La última revisión de esta Shield incluye una ranura para tarjetas microSD, que puede usarse para almacenar archivos a los que se podrá acceder a través de la red. Es compatible con la Arduino UNO y Mega 2560 (utilizando la librería Ethernet que incluye la versión 0019 del software de Arduino). Todavía no se incluye una librería para la tarjeta micro SD en la distribución estándar.



La última revisión de esta Shield también incluye un controlador de reset automático para asegurar que el módulo Ethernet W5100 esté correctamente reseteado al iniciarse. Las revisiones anteriores no eran compatibles con la Arduino Mega 2560 y es necesario resetearlas manualmente después del encendido. La revisión original de la Shield incorporaba una ranura para tarjetas SD de tamaño estándar. Esta ya no es compatible.

Arduino se comunica tanto con un módulo W5100 como con la tarjeta microSD utilizando el bus SPI (a través de la clavija ICSP). Es decir, a través de los pines 11, 12 y 13 de la UNO y los pines 50, 51 y 52 de la Mega 2560. En ambas placas el pin 10 se usa para seleccionar el módulo W5100 y el pin 4 para la tarjeta microSD. Estos pines no pueden ser usados para E/S genéricas. En la Arduino Mega 2560 el pin SS (53) no se utiliza para seleccionar ni el módulo W5100 ni la tarjeta microSD. Este pin SS de la Mega 2560 debe mantenerse como salida o la interfaz SPI no funcionará.

Se debe tener en cuenta que como el módulo W5100 y la tarjeta microSD comparten el bus SPI, solo uno de ellos puede estar activo. En caso se esté usando los dos periféricos en el programa, estos deben ser atendidos por la librería correspondiente. En caso de que se utilice alguno de los dos periféricos es necesario desactivar de forma explícita uno de ellos. Para hacer esto con la tarjeta microSD se debe configurar el pin 4 como salida y escribir HIGH para el pin 4. Para el módulo W5100, configura el pin 10 como salida HIGH.

La Arduino Ethernet Shield proporciona un conector Ethernet RJ45 estándar.

El botón de reset en la Shield resetea tanto el módulo W5100 como la placa Arduino.

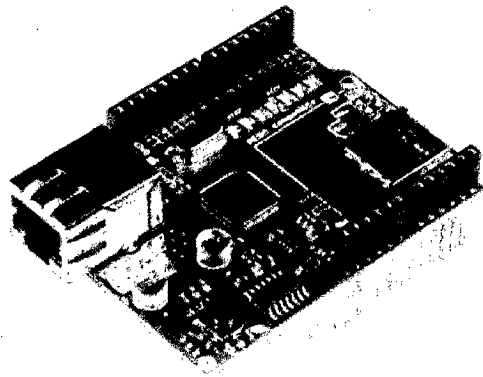


Figura N°22: Shield Ethernet W5100

### **2.3. SISTEMAS DE CONTROL**

Un sistema de control está definido como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado, de modo que se reduzcan las probabilidades de fallos y se obtengan los resultados buscados.

Hoy en día los procesos de control son síntomas del proceso industrial que estamos viviendo. Estos sistemas se usan típicamente en sustituir un trabajador pasivo que controla un determinado sistema (ya sea eléctrico, mecánico, etc.) con una posibilidad nula o casi nula de error, y un grado de eficiencia mucho más grande que el de un trabajador. Los sistemas de control más modernos en ingeniería automatizan

procesos en base a muchos parámetros y reciben el nombre de Controladores de Automatización Programables (PAC).

Los sistemas de control deben conseguir los siguientes objetivos:

- Ser estables y robustos frente a perturbaciones y errores en los modelos.
- Ser eficiente según un criterio preestablecido evitando comportamientos bruscos e irreales. Entre las principales necesidades que nos llevan a realizar la supervisión de procesos, podemos mencionar las siguientes:
- Limitaciones de la visualización de los sistemas de adquisición y control
- Control vs Monitorización • Control software. Cierre de lazo de control
- Recoger, almacenar y visualizar información.
- Minería de datos.

### **2.3.1 Sistemas SCADA**

SCADA quiere decir Supervisory Control And Data Acquisition, y en español significa, Supervisión, Control y Adquisición de Datos y son aplicaciones de software, diseñadas con el objetivo de controlar y supervisar procesos a distancia, proporcionando comunicación con los dispositivos de campo (controladores autónomos, sensores, etc.) y controlando el proceso de forma automática desde una computadora u otro dispositivo. Igualmente, envía la información que se genera en el proceso de producción a uno o

varios usuarios, del mismo nivel y hacia otros entes de supervisión dentro de la empresa, es decir, admite la participación de otras áreas.

En un sistema SCADA se incluyen muchos subsistemas, así la adquisición de los datos puede estar a cargo de un PLC (Controlador Lógico Programable) el cual recibe las señales y las envía a las estaciones remotas usando protocolos determinados, otra manera podría ser a través de una computadora que realice la toma de datos vía hardware especializado y luego transmita la información a un equipo de radio mediante su puerto serial, y así un sin número de alternativas.

Las tareas tanto de supervisión como de control están relacionadas con el software que utilice el sistema SCADA, allí el operador puede visualizar en la pantalla del computador de cada estación remota que conforme el sistema, los estados, las situaciones de alarma y tomar acciones físicas sobre algún equipo lejano, la comunicación se realiza mediante buses especiales o redes LAN. Todo esto se ejecuta normalmente en tiempo real.

Además estos sistemas actúan sobre dispositivos de la planta, y permiten controlar el proceso desde una estación remota. Un software SCADA debe ser capaz de ofrecer al sistema la posibilidad de crear alarmas, que exigen la presencia del operador para reconocer una parada del proceso. Permite la generación de datos históricos de las señale de planta, que pueden ser interpretados en una hoja de cálculo.

También ejecuta programas para anular o modificar las tareas asociadas al proceso, bajo ciertas condiciones y tiene la posibilidad de realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador.

Requerimientos Principales de un SCADA.- Existen varios tipos de SCADA's dependiendo del fabricante y del fin con que se va a hacer uso del sistema, por ello antes de decidir cuál es el más indicado hay que tener presente si cumple o no ciertos requerimientos primordiales:

- Debe ser indispensable que tenga arquitectura abierta, es decir, permita su crecimiento y expansión, también debe poder ajustarse a las necesidades futuras de los procesos de producción y de la planta.

- La programación debe ser sencilla y la instalación debe contar con interfaces gráficas que muestren un esquema básico y real del proceso, pero que además, sean amigables para el usuario.

- Es muy importante que se permita la adquisición de datos de todos los equipos que formen parte del sistema SCADA, y su la comunicación debe ser a nivel interno y externo.

## **CAPITULO III. MATERIALES Y METODOS**

### **3.1 LUGAR DE LA INVESTIGACIÓN.**

La presente investigación se desarrolló en los ambientes del laboratorio de la Escuela de Ingeniería Electrónica y Telecomunicaciones de la Universidad Nacional de Piura, sobre los que se implementa el modelo de monitoreo y control de una planta.

Existe una fuerte tendencia al uso de tecnologías de microcontroladores, placas de desarrollo basadas en Open Source, sensores y módulos Shield que permiten una práctica implementación y desarrollo, no cabe duda que la herramienta ideal para lograr la programación y conectividad sea un computador con alta capacidad de procesamiento, velocidad de respuesta y flexibilidad para establecer sistemas de comunicaciones.



### 3.2 DURACIÓN DE LA INVESTIGACIÓN.

La presente investigación se desarrolló durante los meses de Enero a Mayo del año 2015, a continuación se detallan las actividades realizadas durante la investigación:

ACTIVIDAD / MES	2015				
	Enero	Febrero	Marzo	Abril	Mayo
Recopilación de información	X				
Elaboración de marco teórico	X	X			
Descripción de actividades del sistema		X			
Simulación de hardware y software		X	X		
Implementación de la Planta y Sistema de Monitoreo			X	X	
Elaboración de Informe Final					X

### 3.3 MATERIALES DE LA INVESTIGACIÓN.

Los materiales utilizados en la presente investigación fueron tomados de arquitecturas y tecnologías existentes así como el uso de software libre:

- Computador Core i3 donde se diseñaron las interfaces de usuario final basados en tecnología Web y se realizó la programación de las tarjetas de Arduino y sus respectivos Shield.

- Tarjeta de desarrollo Arduino UNO R3, que fue utilizada para el diseño del prototipo del Sistema de monitoreo y control en tiempo real.
- Tarjeta de desarrollo Arduino MEGA, que fue utilizada específicamente para el módulo de lectura RFID del Sistema de monitoreo y control en tiempo real.
- Módulos Shield Ethernet W5100, que realizaron la comunicación a través de la red que fue implementada.
- Sensores PIR, ultrasónicos, de humedad, temperatura y un módulo RFID para la captura de parámetros, tanto ambientales como de seguridad de la planta que se automatizó.
- Router-Switch con WIFI para la configuración de la topología estrella que se implementó en la red.
- PatchCord's de cable UTP Cat 5E con norma IEEE 568B, para la comunicación de las estaciones hacia el concentrador de red.
- Teléfono celular de la gama de Smartphone con sistema operativo Android para la comunicación y configuración HMI del presente proyecto.
- Cables, herramientas y componentes electrónicos en general

### 3.4 METODOLOGÍA DE LA INVESTIGACIÓN.

Desde el punto de vista de la investigación y basados en los objetivos de éste trabajo, se definió un modelo de investigación estructural de acuerdo al siguiente esquema:

MODELO	VARIABLES	ESTRUCTURA FUNCIONAL
<p>SISTEMA DE SUPERVISION Y CONTROL DE UNA PLANTA APLICANDO COMUNICACIÓN ETHERNET Y TECNOLOGÍA MÓVIL</p>	<p>VI: Comunicación Ethernet y Tecnología Móvil</p> <p>VD: Sistema de Supervisión y Control</p>	<p><math>SSC = M1 + M2 + M3 + M4</math></p> <p>Dónde:</p> <p>SSC = Sistema de Supervisión y Control</p> <p>M1 = Implementación del sistema de adquisición de datos mediante sensores.</p> <p>M2 = Desarrollo del software de los controladores y tarjetas de adquisición.</p> <p>M3 = Definir la arquitectura de la Red.</p> <p>M4 = Diseñar las interfaces de usuario final basado en Web y Android.</p>

## CAPITULO IV. RESULTADOS.

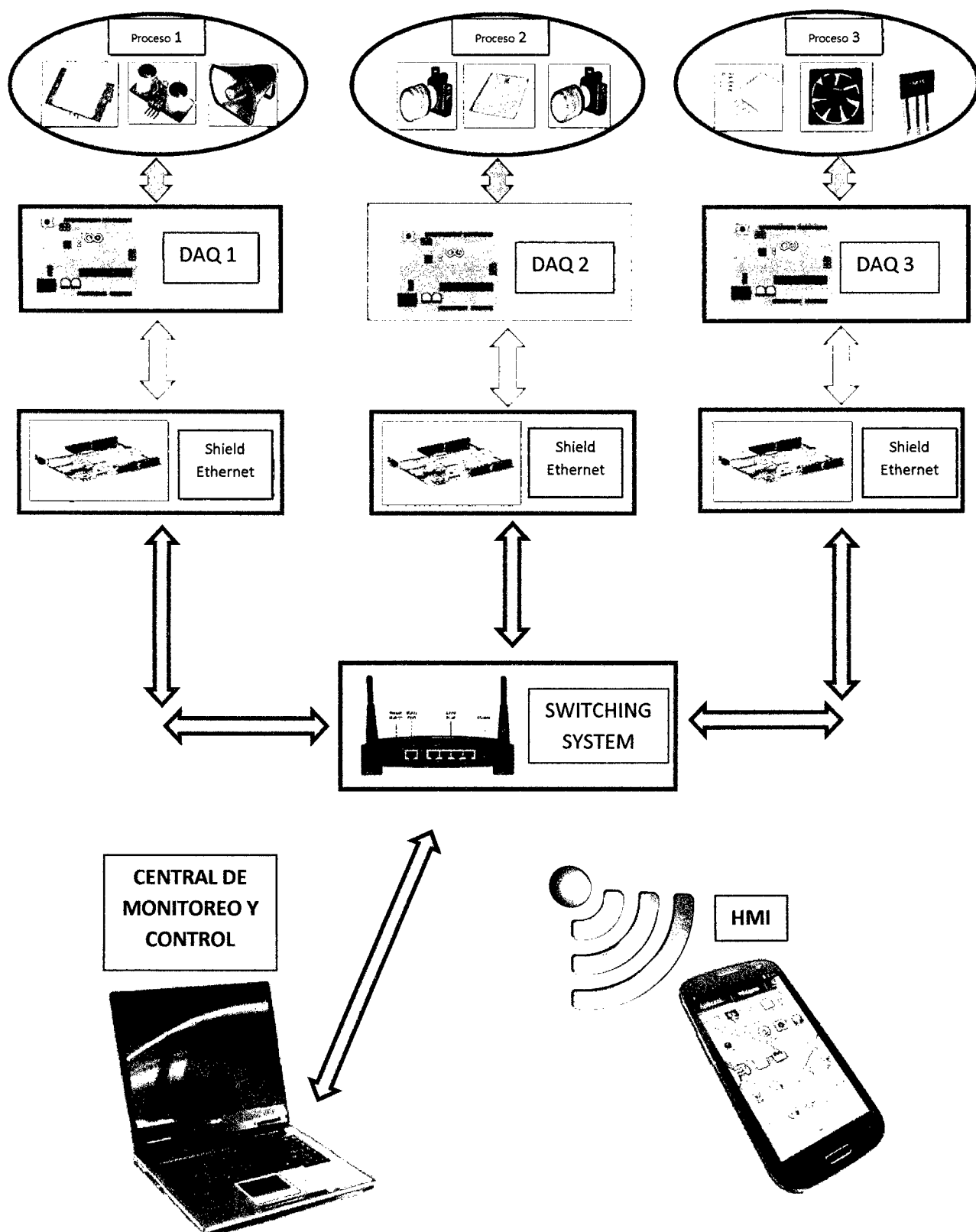


Figura N°23: Diagrama del sistema de supervisión y control

#### **4.1 DESCRIPCIÓN GENERAL DEL SISTEMA.**

En la Figura N°23, se detalla el diseño del sistema propuesto, el cual consiste en el monitoreo y control de 3 procesos independientes:

##### **Proceso 1:**

Módulo de seguridad, está conformado por los sensores de movimiento PIR y el sensor ultrasónico HC-SR04, los cuales se encargan de identificar presencia y proximidad en la parte de seguridad en la planta, como elemento actuador se tiene una alarma sonora que se activará en caso de intrusión u otro evento asignado desde la unidad de mando.

##### **Proceso 2:**

Módulo de identificación y acceso, está conformado por el módulo RFID, que se encargan de registrar el acceso a la planta según el dispositivo de identificación asignado, como elemento visual se cuenta con un piloto led que indicará el correcto acceso al sistema.

##### **Proceso 3:**

Módulo de control de temperatura y humedad, está conformado por los sensores LM 35 y DTH 11, los cuales se encargan de obtener los datos ambientales de la planta, como elemento actuador se tiene un cooler el cual se activará en función al Set Point fijado desde la central o el HMI.

DAQ 1, DAQ 2, DAQ 3:

Es la etapa de adquisición de datos y control de los actuadores, está conformada por una tarjeta de Arduino para cada etapa, lo cual hace que cada etapa funcione de manera independiente y autónoma en función al sistema integrado de supervisión y control.

Shield Ethernet:

Es un módulo Shield W5100 que se encarga de la interfaz entre las tarjetas de Arduino y la comunicación Ethernet, cuenta con un puerto RJ-45 hembra y trabaja a las velocidades convencionales de los dispositivos de red, permite el acople exacto con la DAQ correspondiente y es un dispositivo Web Based que aloja una interface Web para su monitoreo.

Switching System:

Es la etapa que se encarga de interconectar los dispositivos de red, en éste caso cada módulo o proceso que cuenta con la Shield W5100, la unidad central de monitoreo y control y el dispositivo HMI. Además es un dispositivo que cuenta con interconexión WIFI por lo que se pueden conectar y configurar dispositivos móviles.

Central de monitoreo y control:

Ésta etapa está conformada por un computador ya sea una PC de escritorio o Laptop que cuente con comunicación Ethernet y navegador Web, aquí se llevó a cabo el

diseño de las interfaces que posteriormente permiten el monitoreo en tiempo real y el control de todo el sistema desde cualquier punto de la red.

Dispositivo HMI (Interfaz Hombre-Máquina):

En ésta etapa se cuenta con un dispositivo móvil con sistema operativo Android, para el cual se ha diseñado una aplicación desarrollada en App Inventor 2, la cual permite también la operatividad del sistema.

## **4.2 DESARROLLO DEL SOFTWARE DE LAS TARJETAS DE LAS ESTACIONES DEL SISTEMA DE SUPERVISIÓN Y CONTROL.**

### **4.2.1 Módulo de seguridad.**

Para el desarrollo de éste módulo se han empleado sensores de tipo: Ultrasónico y de movimiento PIR, los cuales han sido configurados mediante codificación como las entradas a la tarjeta de adquisición de datos, para éste caso el Arduino UNO R3, las señales provenientes de los sensores son capturadas y enviadas previamente al monitor serial propio de la plataforma de Arduino para su verificación.

A continuación podemos observar la disposición de los sensores en la tarjeta de ésta etapa :

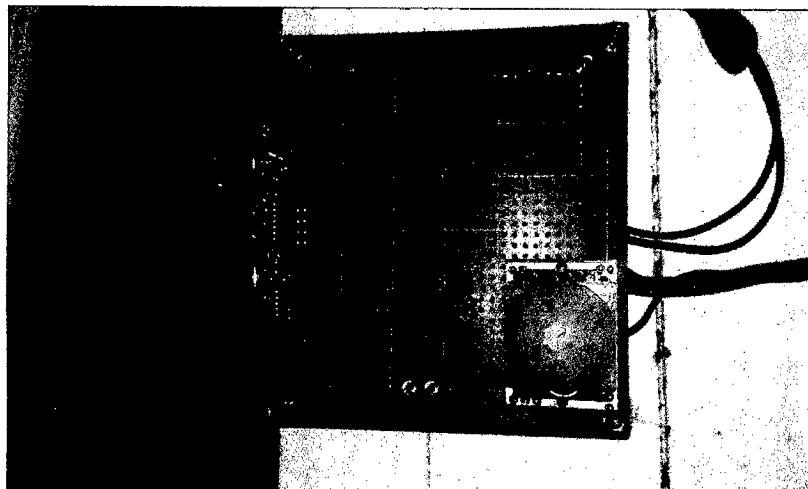


Figura N°24: Vista superior de la placa del módulo de seguridad

Para la etapa de respuesta del procesamiento en las tarjetas se ha instalado dispositivos visuales y sonoros como Diodos LED y una sirena los cuales darán aviso de que se han cumplido las condiciones establecidas en el programa.

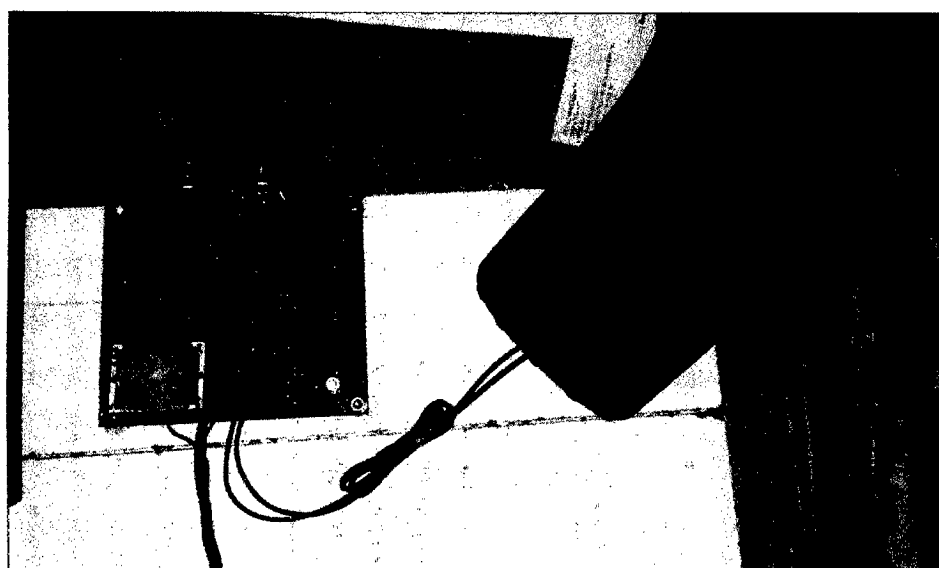


Figura N°25: Vista del módulo de seguridad incorporando sirena



Descripción del funcionamiento del módulo de Seguridad: Se ha programado en el sentido de que se pueda detectar la presencia de personas en un área restringida de una zona de peligro, como ejemplo cercanía a turbinas, calderas, motores, cableado de alta tensión, etc; para demostrar éste evento emite una luz color naranja y el sonido pulsante de la sirena que indica advertencia.

Del mismo modo, en caso de que el trabajador se aproxime demasiado al equipo o maquinaria, se ha ajustado a 30 cm de distancia, el sensor ultrasónico indicará que se active la alarma en forma continua debido a la alerta generada y a la vez se puede visualizar el indicador en rojo.

Finalmente se puede visualizar y monitorear, todo éste proceso vía una interface Web desde la dirección del host: 192.168.1.10

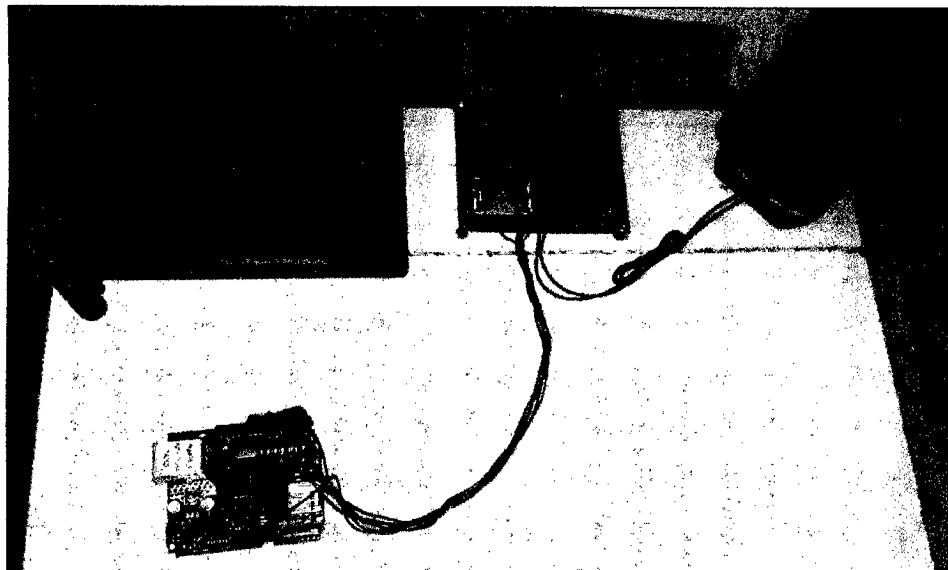


Figura N°26: Módulo de seguridad acoplado al controlador

A continuación se procede a conectar a la red para su comunicación.

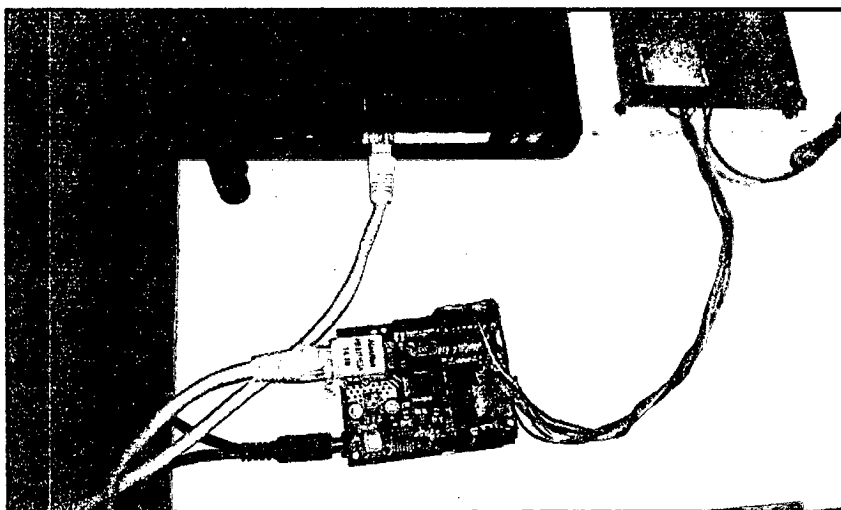


Figura N°27: Conexión Ethernet

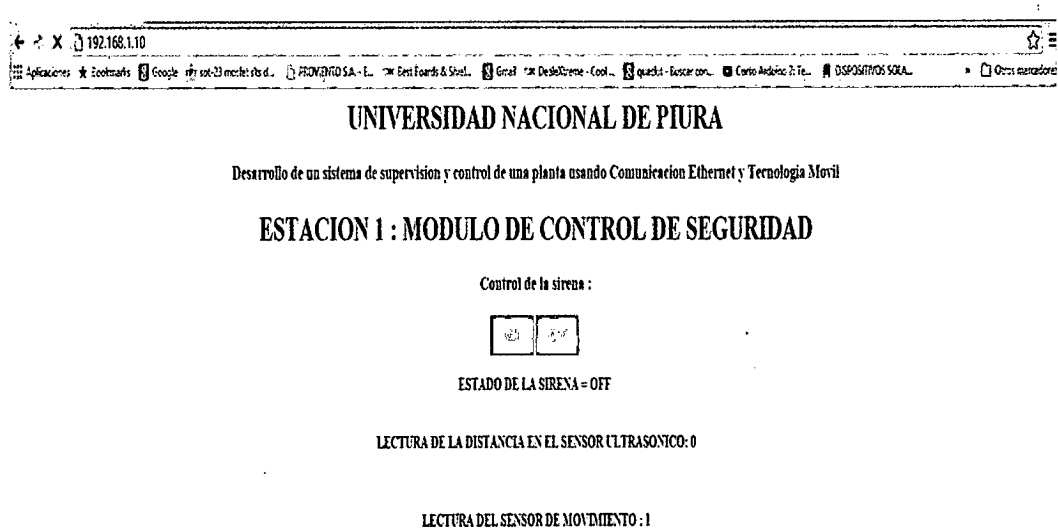


Figura N°28: Interfaz del módulo de seguridad

#### 4.2.2 Módulo de identificación y acceso vía RFID.

Para el desarrollo de éste módulo se han empleado el módulo RFID de modelo RC 522, el cual ha sido configurado para determinar el acceso de trabajadores de una planta u oficina, mediante el código del programa se determina que al pasar los Tags que se han declarado como “permitidos” se obtiene acceso y se identifica con una luz de color verde, caso contrario, el sistema emite una luz color rojo que nos indica acceso denegado, adicionalmente se puede agregar al evento de acceso permitido un actuador que para éste caso puede ser un servo motor que simula el acceso a una tranquera o puerta electrónica.

A continuación podemos observar la disposición del sistema RFID:

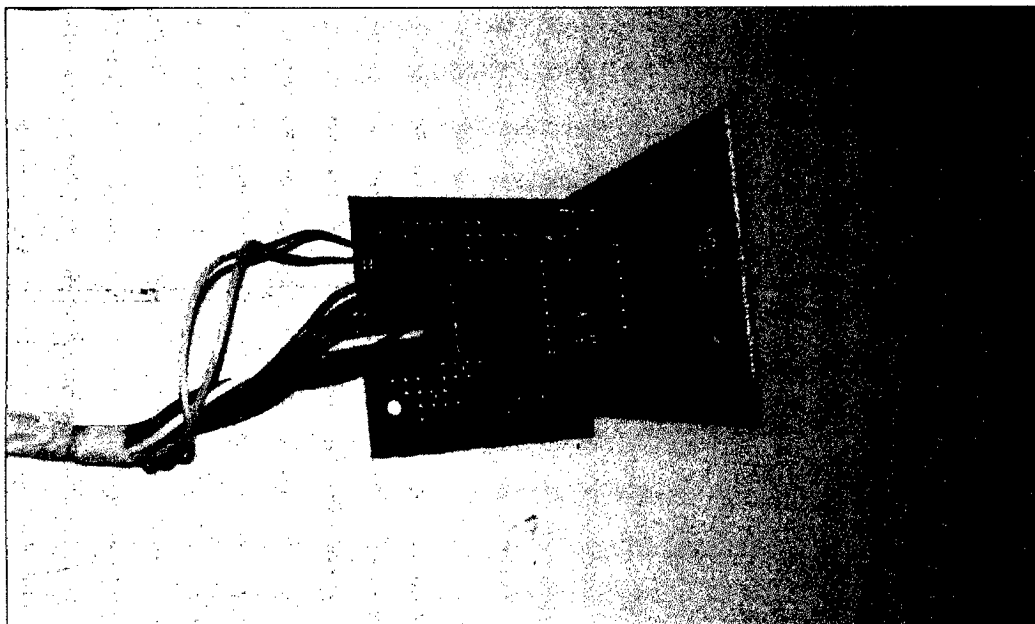


Figura N°29: Vista superior del módulo RFID

Para la etapa de respuesta del procesamiento en las tarjetas se ha instalado dispositivos visuales, como se puede apreciar en la siguiente imagen:

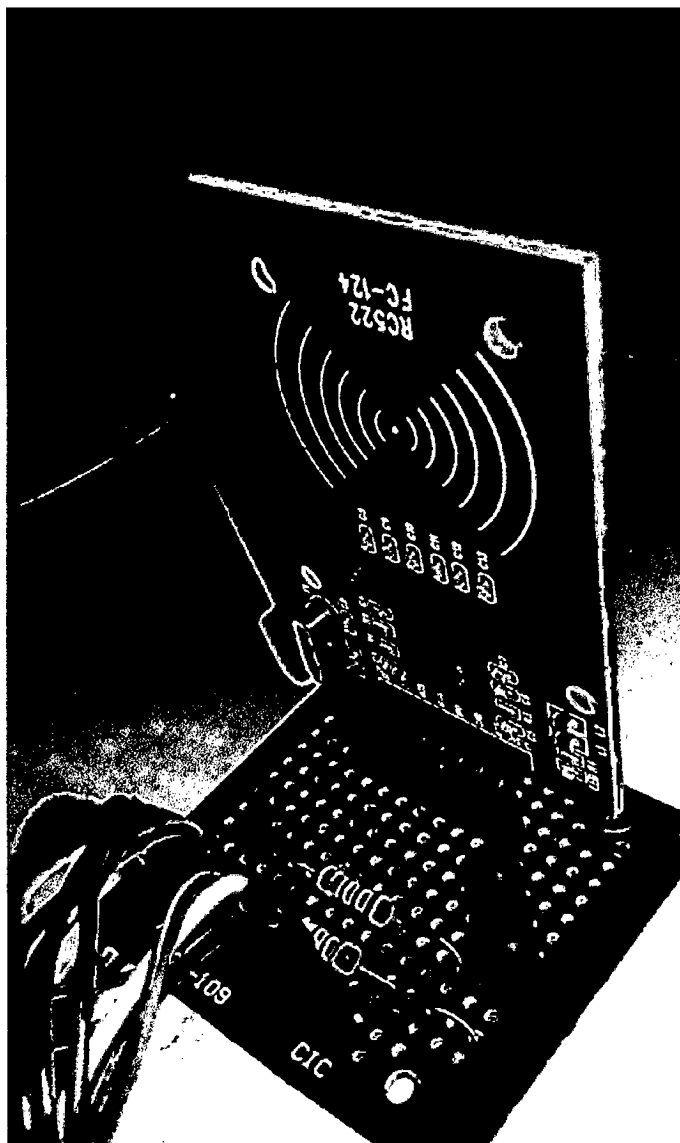


Figura N°30: Indicadores led instalados

Finalmente se puede visualizar y monitorear, todo éste proceso vía una interface Web desde la dirección del host: 192.168.1.20

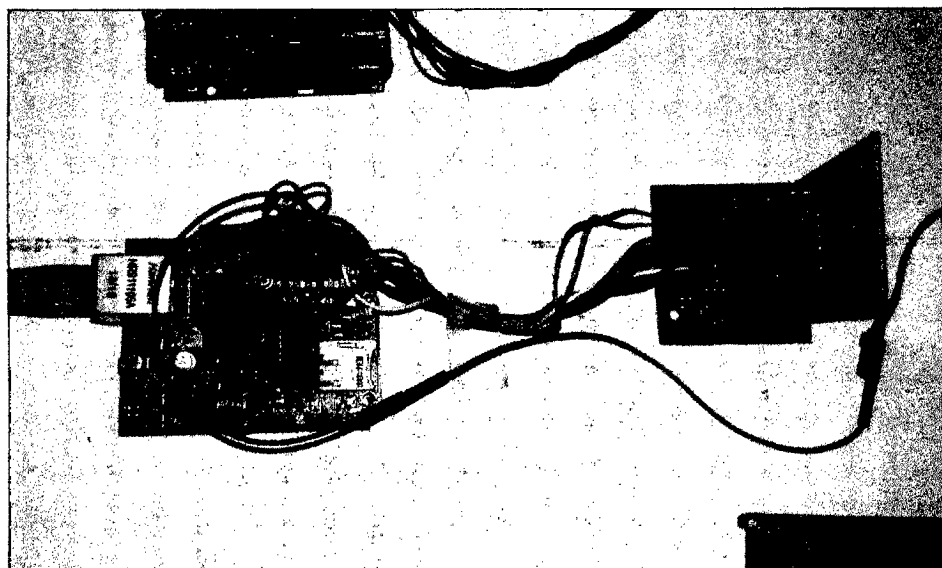


Figura N°31: Módulo de RFID acoplado al controlador

A continuación se procede a conectar a la red para su comunicación.

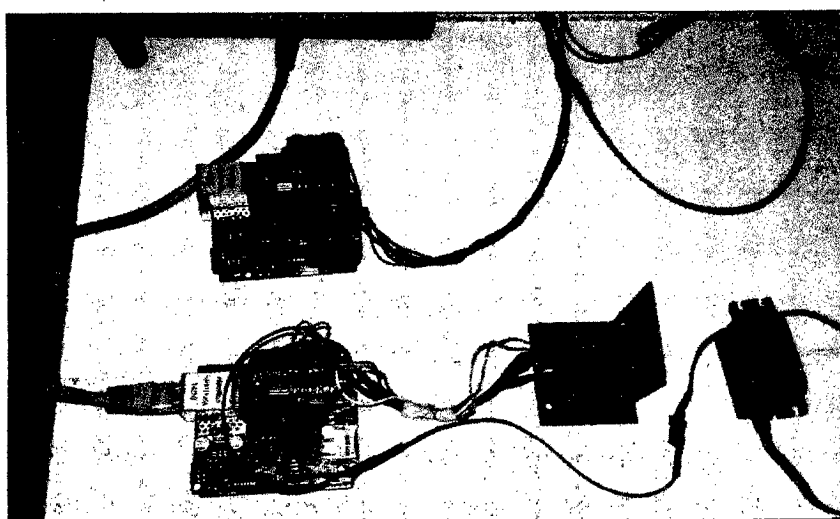


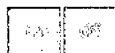
Figura N°32: Conexión Ethernet

## UNIVERSIDAD NACIONAL DE PIURA

Desarrollo de un sistema de supervisión y control de una planta usando Comunicación Ethernet y Tecnología Móvil

### ESTACION 2 : MODULO DE ACCESO VIA RFID

ESTADO DEL SERVOMOTOR :



ESTADO DE LA SIRENA = OFF

LECTURA DEL ACCESO: 0

LECTURA DEL INDICADOR DE ACCESO(LED) : 1

Figura N°33: Interfaz del módulo de acceso vía RFID

#### 4.2.3 Módulo de Control de Temperatura

Para la implementación del módulo de control y monitoreo de Temperatura se ha tenido se han empleado la técnica básica de control de lazo cerrado.

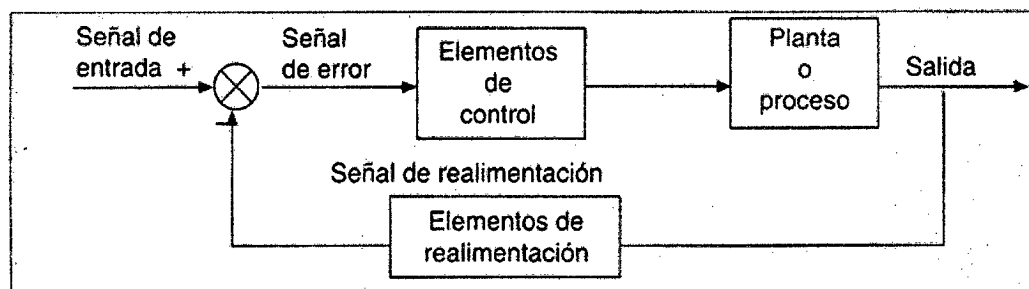


Figura N°34: Diagrama clásico de sistema de control realimentado

Donde :

- Señal de entrada es la entrada al sistema.
- Los elementos de realimentación son los sensores.
- Señal de realimentación es la señal proporcionada por el sensor
- La salida es la variable controlada.
- Señal de error es la diferencia entre la señal de entrada y la señal de realimentación.

Sensores Utilizados: Sensor de temperatura LM 35 y sensor de humedad relativa DHT 11.

Actuador: Cooler de computadora con voltaje de operación de 5Vcc y un consumo máximo de 200mA

A continuación podemos observar la disposición de los sensores en la tarjeta de ésta etapa :

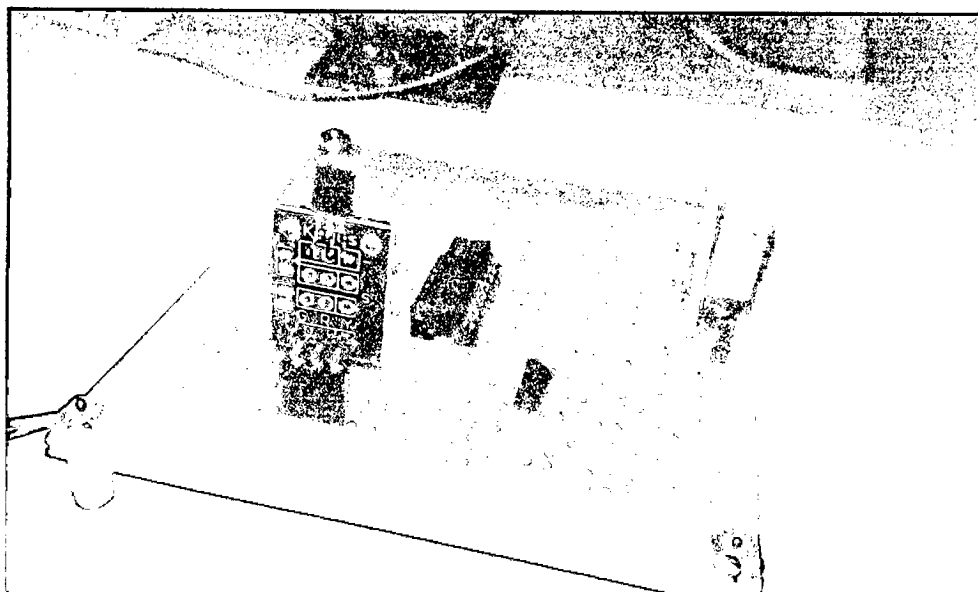


Figura N°35: Vista superior de la placa del módulo de control de temperatura

Para la etapa de respuesta del procesamiento en la tarjeta se ha instalado como elemento actuador, un Cooler que ventilará al sensor de temperatura para efectuar la tarea de control.

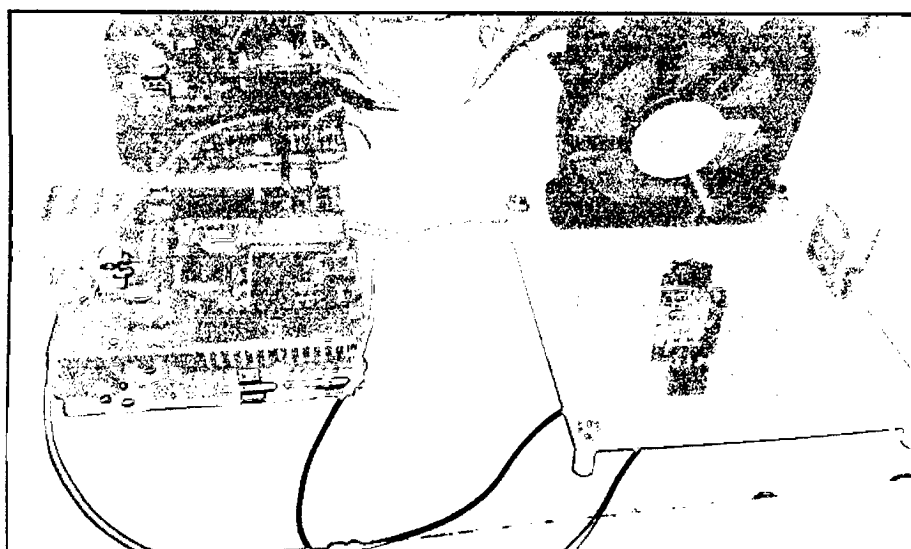


Figura N°36: Vista del módulo de control de temperatura completo



Descripción del funcionamiento del módulo de Temperatura: Se ha programado en el sentido de que se pueda controlar y monitorear la temperatura de la estación 3 desde una interface Web, los sensores instalados pueden recoger las variables ambientales de temperatura y humedad, las cuales serán leídas por la tarjeta de adquisición que en éste caso es un Arduino UNO R3 y según el set point asignado deberá controlar la temperatura que está siendo capturada por el sensor.

Finalmente se puede visualizar y monitorear, todo éste proceso via una interface Web desde la dirección del host: 192.168.1.30

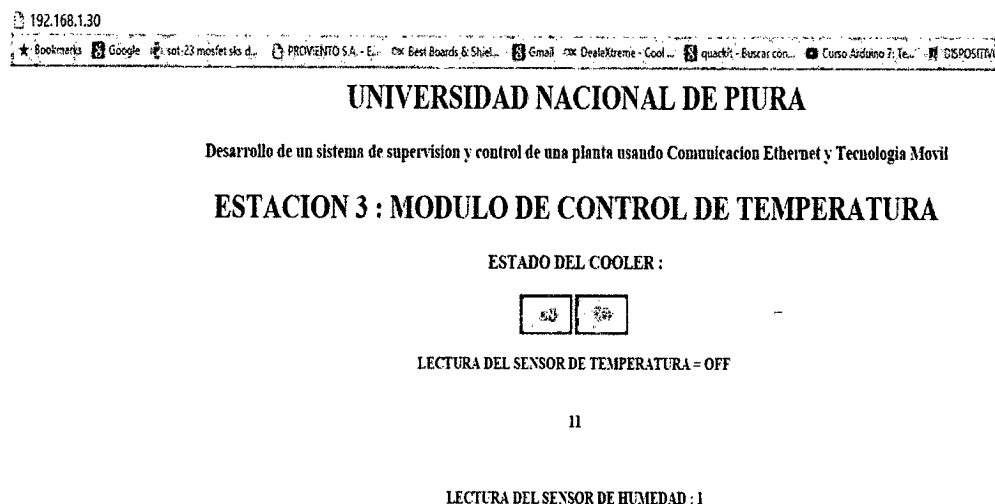


Figura N°37: Conexión Ethernet

#### 4.2.4 Configuración de la red

Para la configuración de la red Ethernet del proyecto, se ha empleado un Router – Switch de la marca Zyxel, modelo P-660HW.

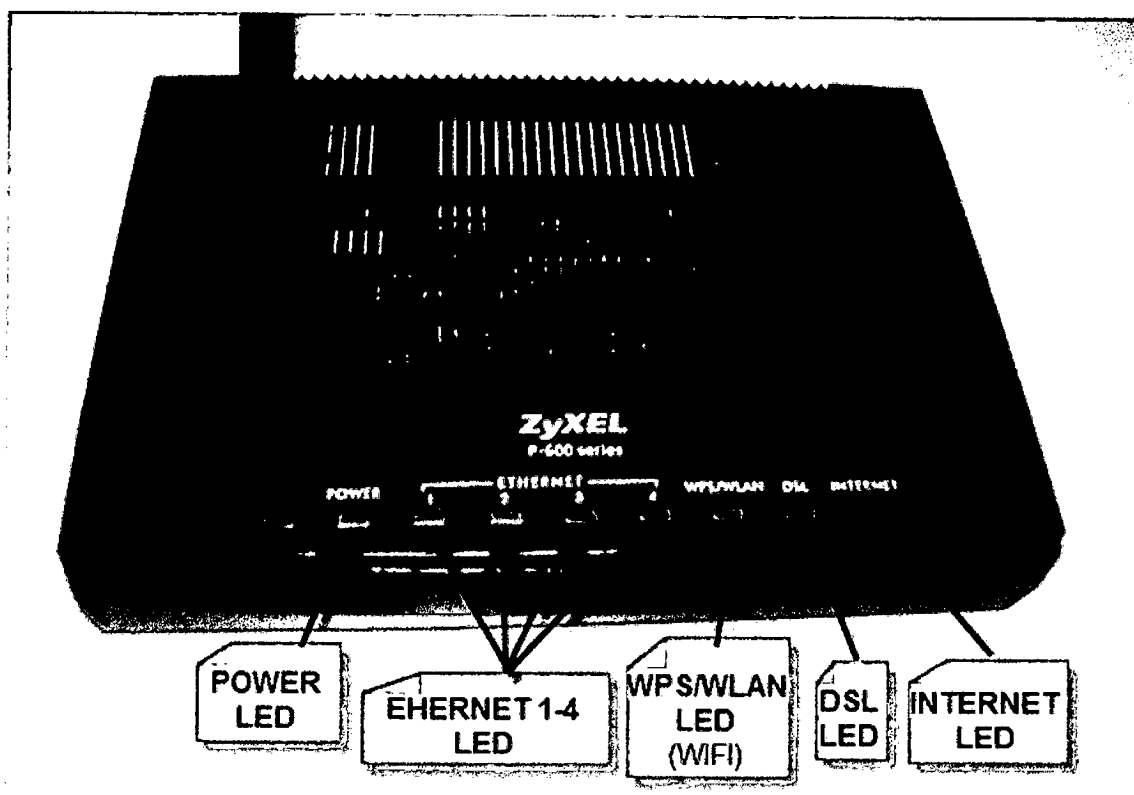


Figura N°38: Vista general del Router P-660HW

Este equipo posee 4 puertos LAN, 1 puerto Módem y tiene capacidad para conexión WIFI, como se muestra en la siguiente figura:

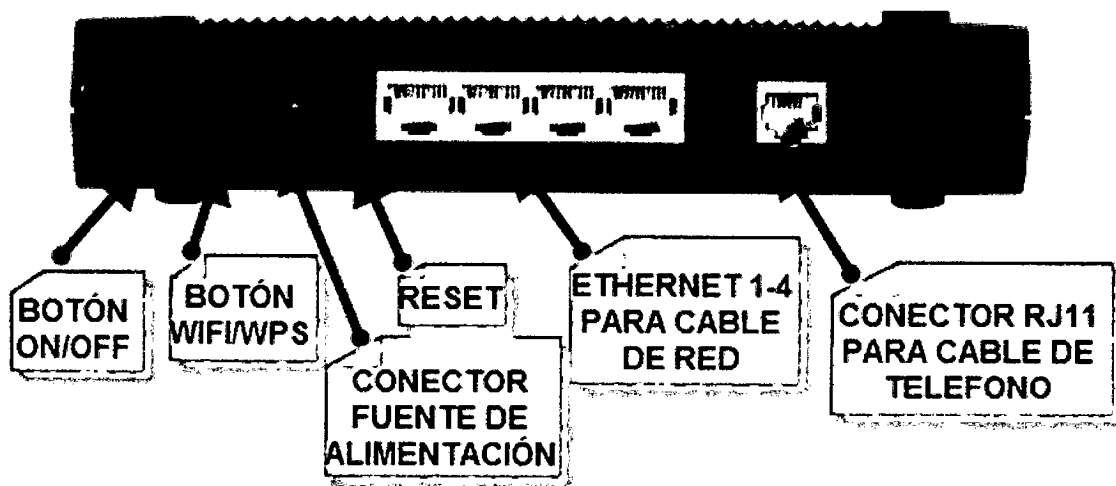


Figura N°39: Vista posterior del Router P-660HW

Los Host o módulos que son parte de la red, poseen una dirección IP estática, se han configurado basados en IP clase C y se han tomado las siguientes IP:

Módulo de Seguridad	:	192.168.1.10
Módulo de control de acceso vía RFID	:	192.168.1.20
Módulo de control de temperatura	:	192.168.1.30
Computador para monitoreo Web	:	192.168.1.50
Dispositivo HMI	:	192.168.1.60

#### 4.2.5 Desarrollo de la aplicación para el Dispositivo HMI

El desarrollo de la Aplicación se ha llevado a cabo en el entorno de App Inventor 2, es un entorno libre, gratuito y basado en desarrollo on-line.

Para empezar a crear aplicaciones en App Inventor 2, sólo es necesario configurar y tener una cuenta de correo electrónico del dominio de Google, es decir que empezamos teniendo una cuenta Gmail, el entorno es totalmente amigable y sencillo y cabe indicar que no es necesario poseer sólidos conocimientos de programación para crear las aplicaciones.

El enlace para acceder es el siguiente:

<http://ai2.appinventor.mit.edu/>

En la siguiente figura, podemos apreciar el entorno general de App Inventor 2:

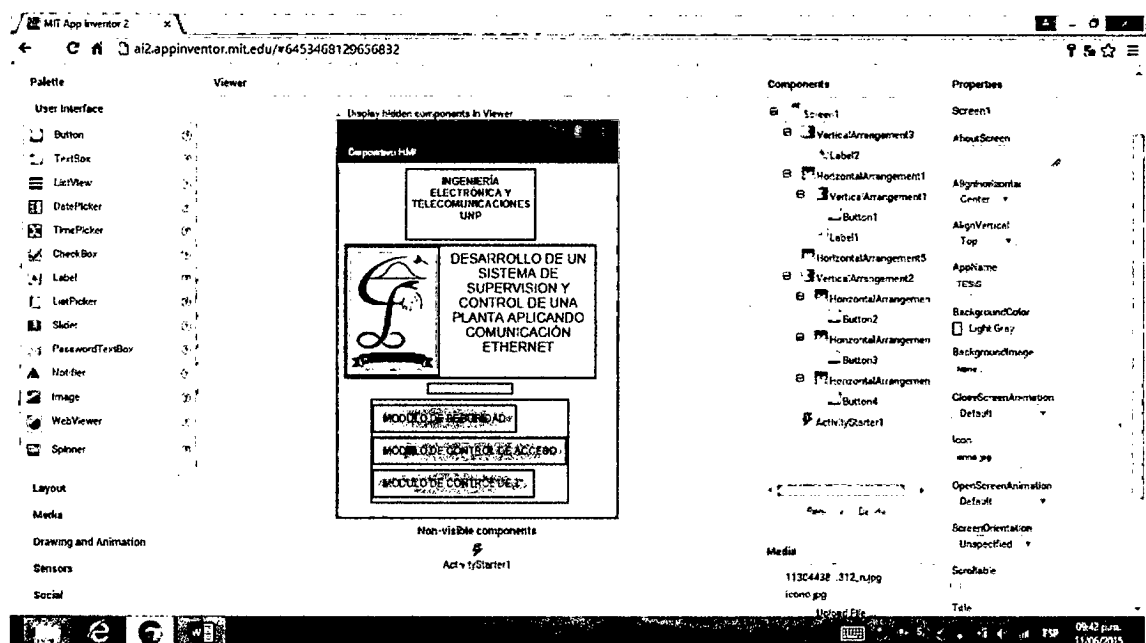


Figura N°40: Entorno Web de App Inventor 2

App Inventor 2 ofrece diversas herramientas de diseño para el desarrollo de aplicaciones, de las cuales se han utilizado las más básicas y conocidas:

a) Button

Los botones son componentes que los usuarios tocan para realizar alguna acción en su aplicación.

Botones pueden detectar cuando el usuario pulse en ellos. Muchos aspectos de la apariencia de un botón se pueden cambiar. Puede utilizar la propiedad Enabled de elegir si un botón puede ser aprovechado.

Propiedades:

1. BackgroundColor:

Color de fondo para el botón.

2. Enabled:

Si se activa, el usuario puede botón de toque para provocar la acción.

3. FontBold

Si se establece, el texto del botón se muestra en negrita.

4. FontItalic

Si se establece, el texto del botón se muestra en cursiva.

5. FontSize

El tamaño en puntos para el texto del botón.

6. FontTypeface

Familia de fuentes para el texto del botón.

7. Height

Altura del botón.

8. Width

Anchura Button.

9. Image

Imagen para mostrar el botón.

10. Text

Texto a mostrar en el botón.

11. TextAlignment

Izquierda, centro o derecha.

12. TextColor

Color para el texto del botón

b) Label

Las etiquetas son componentes que se utilizan para mostrar texto.

Propiedades:

1. BackgroundColor:

Color de fondo para el botón.

2. Enabled:

Si se activa, el usuario puede botón de toque para provocar la acción.

3. FontBold

Si se establece, el texto del botón se muestra en negrita.

4. FontItalic

Si se establece, el texto del botón se muestra en cursiva.

5. **FontSize**

El tamaño en puntos para el texto del botón.

6. **FontTypeface**

Familia de fuentes para el texto del botón.

7. **Height**

Altura del botón.

8. **Width**

Anchura Button.

9. **Image**

Imagen para mostrar el botón.

10. **Text**

Texto a mostrar en el botón.

11. **TextAlignment**

Izquierda, centro o derecha.

12. **TextColor**

Color para el texto del botón.

13. **Visible**

Si se establece, la etiqueta es visible.

c) **HorizontalArrangement**

Se utiliza para alinear y apilar componentes como los botones, etiquetas, cuadros de imágenes, etc. de manera horizontal.

d) **VerticalArrangement**

Se utiliza para alinear y apilar componentes como los botones, etiquetas, cuadros de imágenes, etc. de manera vertical.

Finalmente se realiza la programación en bloques desde el App Inventor 2, para efectos del presente proyecto, necesitamos que la aplicación posea botones que puedan abrir direcciones Web específicas que corresponden a los host desde donde se ejecutan las interfaces Web de los módulos o estaciones presentes en la red.

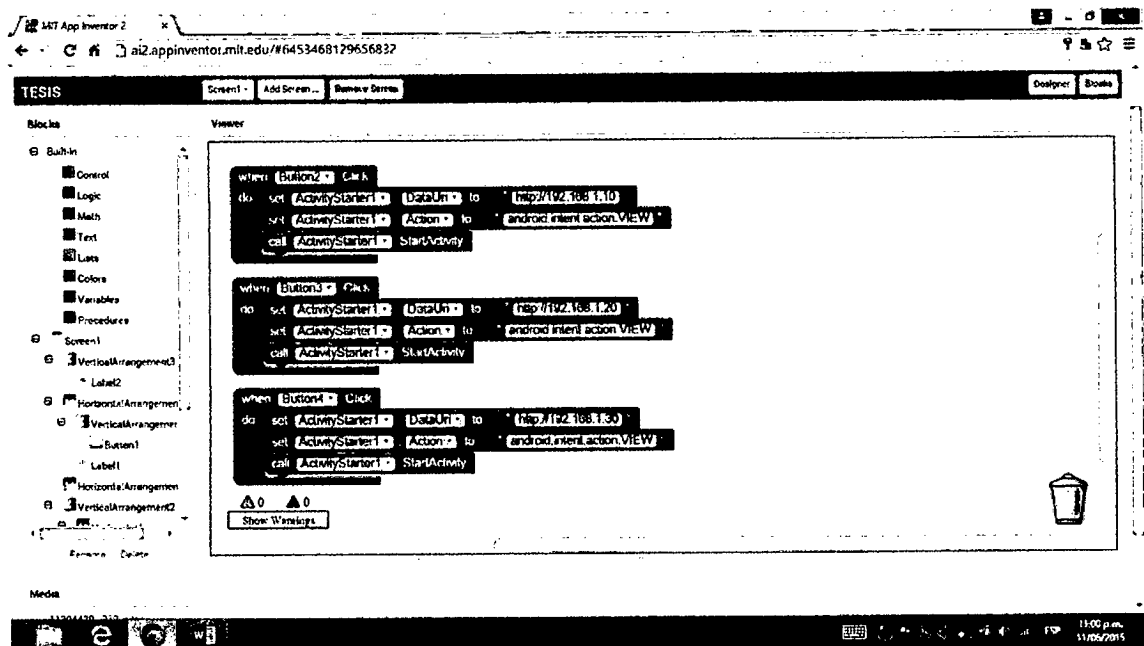


Figura N°41: Programa en bloques de la aplicación



## CAPITULO V. ANALISIS Y DISCUSIÓN

A continuación vamos a presentar el análisis del desarrollo del presente proyecto, conforme se fue desarrollando pudimos comprobar teorías obtenidas durante el proceso de aprendizaje en los estudios de pregrado en nuestra gloriosa alma mater de Piura. Debemos indicar que durante la implementación de los dispositivos integrados en el proyecto, tales como las tarjetas Arduino, sensores y la red, hubo contratiempos técnicos que fueron superados en la etapa de experimentación y pruebas.

Los módulos implementados fueron 3, el primero corresponde a un sistema de control de proximidad para Seguridad ocupacional; en ésta etapa inicial del proyecto, se tuvo que realizar una calibración previa de los sensores tanto el de distancia como el de movimiento, en el caso del segundo, la calibración fue de manera física dando un pequeño ajuste al potenciómetro integrado del PIR, una vez hecho esto se procedió a dar lectura serial de ambas variables para comprobar la conectividad con la tarjeta, luego de implementado y probado se procede a tomar las medidas del tamaño de programa para determinar su rendimiento.

### **Módulo de Seguridad:**

Tarjeta Base:	Arduino UNO R3
Controlador:	AtMega 328
Capacidad de memoria flash:	32KB
Tamaño de Programa:	7.79 KB (7,986 bytes)
Porcentaje de uso de memoria:	24%

Shield:	Ethernet W5100
Velocidad de Transferencia:	10/100 MBPS
Conectores de red:	RJ-45
Standard de cableado:	Patch Cord 568 – B
Dirección IP:	192.168.1.10
Interface:	Web Based
Tiempo de Actualización:	500 m seg
Tiempo de respuesta:	2 seg

Para la implementación del segundo módulo que corresponde al control de acceso vía RFID, se tuvo que realizar un cambio de tarjeta de adquisición puesto que según las prueba realizadas se determinó que la tarjeta Arduino UNO R3, no trabaja adecuadamente cuando se ha ocupado el 60% de la memoria del microcontrolador, debemos tener en cuenta que la plataforma de Arduino incluye su propio bootloader y por tanto, debemos asumir el tamaño de ese pequeño programa. Se tuvo que realizar un cambio de Arduino, del UNO R3 al MEGA, puesto que para manejar RFID necesitamos sus propias librerías y eso agrega datos al programa que suman en cuanto a tamaño, luego de implementado y probado se procede a tomar las medidas del tamaño de programa para determinar su rendimiento.

### **Módulo de control de acceso vía RFID:**

Tarjeta Base:	Arduino MEGA
Controlador:	AtMega 2560
Capacidad de memoria flash:	256KB
Tamaño del bootloader:	8 KB
Tamaño de Programa:	16 KB
Porcentaje de uso del programa:	6.25%
Porcentaje total de uso: (programa + bootloader)	9.37%
Shield:	Ethernet W5100
Velocidad de Transferencia:	10/100 MBPS
Conectores de red:	RJ-45
Standard de cableado:	Patch Cord 568 – B
Dirección IP:	192.168.1.20
Interface:	Web Based
Tiempo de Actualización:	500 m seg
Tiempo de respuesta:	2 seg
Actuador adicional:	Servomotor pequeño

Finalmente, para la implementación del tercer módulo que corresponde al control de temperatura, se ha implementado la técnica de control ON – OFF, que consiste básicamente en comparar un Set Point previamente ajustado con el valor devuelto por los sensores, se han utilizado para estos fines : 01 sensor de temperatura y 01 sensor de humedad, ambos pueden conectarse directamente a la placa de Arduino pero para lograr equivalencia entre las lecturas, se puede implementar un acondicionamiento de señal para el LM – 35 que consiste en un amplificador no inversor de ganancia 5. Para éste caso no hubo necesidad de calibración previa de los sensores pues son directamente compatibles con Arduino UNO R3, a diferencia del módulo antes expuesto, no hubo problemas de saturación o ineficiencia de trabajo de la memoria operativa del sistema implementado.

#### **Módulo de control de temperatura:**

Tarjeta Base:	Arduino UNO R3
Controlador:	AtMega 328
Capacidad de memoria flash:	32 KB
Tamaño de Programa:	7.5 KB
Porcentaje total de uso:	23.43%
Shield:	Ethernet W5100
Velocidad de Transferencia:	10/100 MBPS
Conectores de red:	RJ-45

Standard de cableado:	Patch Cord 568 – B
Dirección IP:	192.168.1.30
Interface:	Web Based
Tiempo de Actualización:	500 m seg
Tiempo de respuesta:	2 seg
Actuador adicional:	Cooler ventilador

#### La Red:

Con respecto a la red implementada, hemos basado la comunicación en una red Ethernet LAN ya que es la más factible de implementar y además la variedad de dispositivos de redes basados en el protocolo TCP-IP, es cada vez más amplia, se pueden encontrar Switch's, Routers, adaptadores de WLAN, etc. De diversas marcas, modelos y funcionalidad según la necesidad, en nuestro caso bastó con 4 puertos LAN ya que se tienen 3 módulos de procesos y un computador conectado a la red para el constante monitoreo, además el equipo utilizado posee conexión Wi Fi, lo que facilita poder agregar dispositivos inalámbricos como Laptop, Tablet, Smartphones, etc. En resumen, hemos demostrado que la implementación de una red para comunicación y control de procesos es totalmente factible y permite soluciones rápidas que pueden ayudar y contribuir al desarrollo de modelos más avanzados y robustos que se puedan utilizar a nivel industrial.

La topología de la red juega un papel importante en el diseño ya que sobre ella gira el fundamento de la comunicación Ethernet, específicamente hemos empleado la topología Estrella con IP de clase C y obviamente el rango es básico y corto puesto que para efectos de demostración se han empleado 5 IP para 5 Host en los cuales se ha logrado obtener una comunicación eficaz que permite comparar la comunicación de nuestra red con los sistemas de comunicación industrial más conocidos como lo son el Mod Bus, RS-485, Profi Bus, etc.

Con respecto a la parte inalámbrica, podemos conectar cualquier dispositivo que posea comunicación Ethernet, debiendo tener en cuenta que el desarrollo de la aplicación implica un archivo específico de instalación por lo que el dispositivo móvil a usar deberá poseer sistema operativo Android.

#### La Aplicación:

Con respecto al desarrollo de la aplicación podemos indicar varios puntos ya que nos encontramos con un entorno muy novedoso pero bastante amigable y un nivel de programación muy alto lo cual permite la creación de aplicaciones según la necesidad, para nuestro caso se necesitó hacer una aplicación que sea capaz de direccionar o abrir una página Web, debemos indicar que la página Web en referencia es la interface propia que hemos creado y programado en cada uno de los módulos.

El entorno de App Inventor 2 permite generar el archivo de instalación que va a ser almacenado en el Smartphone, se puede acceder a dicho archivo de dos maneras, la primera opción es descargar a la PC el archivo y luego copiarlo en la memoria para ser

abierto durante su instalación, la segunda opción es poder descargar directamente desde internet por medio del AppStore de Android mediante un código QR que fácilmente puede ser leído por una aplicación de lectura de códigos QR como por ejemplo el QR-Droid y así obtener el enlace de descarga directa, almacenar y proceder a instalar la aplicación a ser usada.

Algo importante que debemos acotar es acerca de la extensión del archivo de instalación que nos brinda App Inventor 2, y es un archivo de formato .Apk, éste tipo de extensión del archivo es propio y corresponde al archivo instalador de la plataforma de los sistema operativos Android, es decir, la aplicación que hemos desarrollado puede ser instalada y/o configurada desde cualquier Smartphone o Tablet que tengan Android con sistema operativo nativo.

Además pudimos observar que existen en la red y sobre todo en el store de aplicaciones del propio Google, miles de aplicaciones similares que pueden ser utilizadas para pruebas, la desventaja de esto es que no podemos editar nada del programa, es decir su uso se limita a pruebas.

## CONCLUSIONES.

Después de haber desarrollado e implementado ésta tesis, podemos concluir en que se deben tener en cuenta diversos aspectos técnicos al momento de la instrumentación, los cuales hemos podido documentar en función de las etapas descritas en la metodología del proyecto.

- 1- Vamos a empezar describiendo las conclusiones en cuanto al SSC = Sistema de Supervisión y Control, las cuales nos llevan a indicar que se ha logrado la comunicación Ethernet para implementar una red de procesos ya sean domóticos, electrónicos, mecánicos, e incluso industriales; dicha comunicación ha resultado muy eficiente y demostramos que es bastante práctica de implementar.

Además se ha podido demostrar que el uso de plataformas libres como lo es Arduino, facilita el desarrollo de diversas aplicaciones y en conjunto con la parte de Software como al App Inventor 2 hacen de éstas herramientas un entorno muy completo y versátil al momento de investigar.

- 2- En cuanto a M1 = Implementación del sistema de adquisición de datos mediante sensores; podemos concluir en que actualmente se dispone de una amplia gama de transductores que nos permiten tener más opciones de instrumentación en los proyectos electrónicos. Es importante indicar que también se dispone de diversa información digital en la internet, la



cual nos amplía el panorama al momento de la selección de los sensores en función de la necesidad y la aplicación a desarrollarse.

La etapa de la red de sensores nos permitió practicar mucho con ellos, ya que hubo necesidad de calibrar y observar su comportamiento ante factores externos a su modo o ambiente convencional de trabajo.

- 3- Continuamos con M2 = Desarrollo del software de los controladores y tarjetas de adquisición, de ésta etapa concluimos en que la plataforma de Arduino tanto en software como en hardware permite una gran facilidad de programación y análisis de errores al momento de compilar un programa.

El lenguaje de Arduino es un lenguaje propio y se parece mucho a Java y Processing por la forma de su sintaxis, pero el hecho de que la tarjeta de Arduino posea su propio bootloader, lo hace un entorno Plug and Play y que ha sido parte fundamental para llevar a cabo el presente trabajo.

- 4- Para la parte M3 = Definir la arquitectura de la Red, se hizo un análisis rápido para comparar las distintas topologías de redes que se pueden implementar y se llegó a la conclusión que la topología estrella es la más apropiada porque se amolda a la necesidad de comunicación que fue efecto de la investigación. La topología estrella es muy usada actualmente en las redes de comunicación de datos, debido a su difusión,

también se puede encontrar amplia cantidad de bibliografía como por ejemplo el manual de instrucción CCNA que se basa en la certificación de profesionales para la marca Cisco.

Además se cuenta con información en la internet que hace más flexible la búsqueda y decisión de implementación, para nuestro proyecto se pudo probar y comparar las topologías Anillo y Estrella, pero como incluimos un dispositivo inalámbrico, la segunda opción fue la más adecuada.

- 5- Finalmente en M4 = Diseñar las interfaces de usuario final basado en Web y Android, se tuvo en cuenta de que al acoplar una Shield de tipo Ethernet al Arduino, se puede obtener un host que puede alojar una interface web la cual puede ser monitoreada constantemente desde cualquier navegador, la codificación de las interfaces web se basó en html.

Para el desarrollo del código en html se ha usado un editor en línea que permite bastante facilidad al momento de programar las páginas web que se ejecutan desde cada host.

En cuanto al desarrollo de la aplicación en Android concluimos que se pueden hacer versiones cada vez más complejas y a la vez optimizar los recursos para obtener mejor desempeño del sistema, El desarrollo es gratuito, por tanto su revisión y optimización son totalmente factibles.

## **RECOMENDACIONES.**

Al concluir con el desarrollo de nuestra tesis, podemos emitir las recomendaciones siguientes:

- 1- La principal recomendación es investigar más y profundizar en el desarrollo de aplicaciones basadas en Hardware y Software libre ya que existe una inmensa comunidad de trabajo colaborativo en internet y que no cuesta nada acceder a ella, además es crucial indicar que al usar éstas plataformas no caemos en delitos de usar plataformas propietarias por las que obligatoriamente deberíamos adquirir licencias para su uso y desarrollo de aplicaciones.
- 2- En cuanto a la gama de Arduino disponibles en el mercado, todos son muy buenos y eficientes y la probabilidad de fallas en su hardware es bastante baja, nuestra recomendación va en cuanto al modelo de placa de Arduino a seleccionar ya que según la experiencia obtenida en éste trabajo pudimos comprobar que al usar el 60% de la memoria flash del controlador, el rendimiento se ve afectado considerablemente, por lo que se debe verificar el tamaño de programa que se ha hecho antes de adquirir o determinar la placa de Arduino a utilizar.
- 3- Verificar siempre la corriente de consumo de los dispositivos incluidos como cargas en nuestro sistema, si bien es cierto, Arduino tiene un buen desempeño, hay que tener en cuenta de que la corriente de salida de sus puertos es limitada y

baja, por tanto se debe verificar éste parámetro para evitar corto circuitos y sobre todo dañar la placa por la exigencia de corriente, esto se puede solucionar fácilmente con un adecuado acondicionamiento de nuestras salidas en los puertos de escritura del Arduino.

- 4- Es de vital importancia para ingenieros e investigadores el uso de tecnología y herramientas de software moderno como los que hemos expuesto en la presente tesis, que les permita el análisis y procesamiento de información de forma rápida y concisa, para obtener de este modo, aplicaciones y soluciones prácticas y eficientes que faciliten cada vez más la implementación y desarrollo de nuevas tecnologías aplicada al monitoreo y control de procesos.
- 5- Los sistemas domóticos e industriales de control en la actualidad están siendo reemplazados por equivalentes que pueden operar mayor cantidad de información en menor tiempo, la aplicación y combinación de Arduino y Android, resulta un excelente binomio de desarrollo por lo que la implementación y comparación entre éstas plataformas tiene gran importancia para comprender, evaluar y experimentar las diferentes ventajas que nos ofrecen.

## REFERENCIAS BIBLIOGRAFICAS.

1. Samuel M.(2012). Tesis: Diseño e implementación de un sistema ultrasónico de ayuda para parqueo de vehículos automotrices de un banco de prueba para escuela de ingeniería Automotriz, Escuela Superior Politécnica de Chimborazo, Riobamba-Ecuador.
2. Xicohténcatl C.(2011). Tesis: Sistema de proximidad ultrasónico. Instituto Politécnico Nacional de México DF, Escuela Superior de Ingeniería Mecánica – Eléctrica, Mexico DF.
3. Emilio LL. (2012). Tesis: Diseño de un sistema de control domótico basado en la plataforma Arduino. Escuela Técnica Superior de Ingeniería Informática, Universidad Politécnica de Valencia, España.
4. Hernández, Roberto, Fernández, Carlos y Baptista, Pilar. Metodología de la Investigación. 3a ed., México, Ed. McGraw-Hill, 2003. OLLERO BATURONE, Aníbal, Robótica manipuladores y robots móviles, 1<sup>ra</sup> Edición, Editorial MARCOMBO S.A., Barcelona-España, 2001
5. JORDA, Pedro, y otros, Diseño e implementación de programas en lenguaje c, 1<sup>ra</sup> Edición, Editorial SERVICIO DE PUBLICACIONES, Valencia-España, 1998
6. CRAIG, John, Robótica, 3<sup>ra</sup> Edición, Editorial PEARSON, México, 2000.

7. MANDADO, Enrique, y otros, MicrocontroladoresPic, 1<sup>ra</sup> Edición, Editorial MARCOMBO S.A., Barcelona-España, 2007
8. VALDÉS, Fernando y PALLÀS, Ramón, Microcontroladores: Fundamentos y aplicaciones con PIC, 3<sup>ra</sup> Edición, Editorial MARCOMBO S.A., Barcelona-España, 2007.
9. [www.arduino.cc](http://www.arduino.cc)

## ANEXOS

### **CODIGO EN ARDUINO DEL MODULO DE SEGURIDAD:**

```
#include<SPI.h>//Importamos librería comunicación SPI para el uso del puerto ICSP

#include<Ethernet.h>//Importamos librería Ethernet de la Shield W5100

//Configuración de la red:

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };//Ponemos la dirección MAC
de la Ethernet Shield

IPAddressip(192,168,1,10); //Asignamos la IP al Arduino

EthernetServer server(80); //Creamos un servidor Web con el puerto 80 que es el puerto
HTTP por defecto

//Configuración para la lectura de movimiento:

intledPIR = 7;           // Led que indica presencia en el PIR (Led naranja)

intinputPIR=2;           // Pin digital para lectura del sensor PIR

int movimiento = LOW;    // condición inicial de la variable: movimiento

//Configuración para la lectura de distancia:

intval = 0;

long distancia;

long tiempo;

intsirena=5;

intledsirena=4; // Led que indica proximidad menor a 30 cm en el sensor ultrasónico
(Led rojo)

intrigger=9;           // Puerto por donde enviamos la señal de transmisión del pulso
```

```

ultrasónico

int echo=8;      // Puerto por donde recibimos la respuesta del pulso ultrasónico

String estado="OFF"; //Estado de la sirena inicialmente "OFF"

void setup()

{

//Declaración del modo de operación de los pines del PIR:

pinMode(ledPIR, OUTPUT);    // declaramos el led Naranja como salida

pinMode(inputPIR, INPUT);   // declaramos la entrada del sensor PIR

//Declaración del modo de operación de los pines del Ultrasonido:

pinMode(sirena, OUTPUT);

pinMode(ledsirena, OUTPUT);

pinMode(trigger, OUTPUT); //activación del pin 9 como salida: para el pulso
ultrasónico

pinMode(echo, INPUT); //activación del pin 8 como entrada: tiempo del rebote del
ultrasonido*/

Serial.begin(9600); //sincronizamos la velocidad del puerto serial: 9600 baudios para el
módulo Arduino UNO

//Inicializamos la comunicación Ethernet y el servidor

Ethernet.begin(mac, ip);

server.begin();

Serial.print("El servidor se inicializa con la IP: ");

Serial.println(Ethernet.localIP());

pinMode(orden, OUTPUT);

}

```



```

voidloop()

{

//código de operación del Ultrasonido

digitalWrite(trigger,LOW); // Por cuestión de estabilización del

sensordelayMicroseconds(5); // se ha establecido el tiempo en microsegundos

digitalWrite(trigger, HIGH); // enviamos del pulso ultrasónico

delayMicroseconds(10);

tiempo=pulseIn(echo, HIGH); /* Función para medir la longitud del pulso entrante.

Mide el tiempo que transcurre entre el envío del pulso ultrasónico y cuando el sensor

recibe el rebote*/

distancia= int(0.017*tiempo); /*fórmula para calcular la distancia obteniendo un valor

entero*/

/*Se visualiza la distancia en centímetros por el monitor serial*/

Serial.println("Distancia ");

Serial.println(distancia);

Serial.println(" cm");

delay(500);

if (distancia <= 30)

{

digitalWrite(sirena, HIGH);

digitalWrite(ledsirena, HIGH);

delay(3000);

digitalWrite(sirena, LOW);

digitalWrite(ledsirena, LOW);

```

```

    }

    //código de operación del sensor PIR

    val = digitalRead(inputPIR); // Leemos el pin del PIR, es el pin2

    if (val == HIGH) // En caso el PIR esté detectando movimiento :
    {
        digitalWrite(ledPIR, HIGH);

        delay(100);

        digitalWrite(ledPIR, LOW);

        delay(100);

        digitalWrite(ledPIR, HIGH);

        delay(100);

        digitalWrite(ledPIR, LOW);

        delay(100);

        if (movimiento == LOW)
        {
            movimiento = HIGH;

        }
    }

    else
    {
        digitalWrite(ledPIR, LOW);

        if (movimiento == HIGH)
        {

```

```

    movimiento = LOW;

}

}

//Programación del Módulo Ethernet:

EthernetClient client = server.available(); //Creamos un cliente Web

//Cuando detecte un cliente a través de una petición HTTP

if (client) {

    Serial.println("new client");

    boolean currentLineIsBlank = true; //Una petición HTTP acaba con una línea en
    blanco String cadena=""; //Creamos una cadena de caracteres vacía

    while (client.connected()) {

        if (client.available()) {

            char c = client.read(); //Leemos la petición HTTP carácter por carácter

            Serial.write(c); //Visualizamos la petición HTTP por el Monitor Serial

            cadena.concat(c); //Unimos el String 'cadena' con la petición HTTP (c). De esta manera
            convertimos la petición HTTP a un String

            //Ya que hemos convertido la petición HTTP a una cadena de caracteres, ahora
            podremos buscar partes del texto.

            int posicion=cadena.indexOf("LED="); //Guardamos la posición de la instancia
            "LED=" a la variable 'posicion'

            if(cadena.substring(posicion)=="LED=ON") //Si a la posición 'posicion' hay "LED=ON"

                {

                    digitalWrite(sirena,HIGH);

                    estado="ON";

```

```

    }

    if(cadena.substring(posicion)=="LED=OFF")//Si a la posición 'posicion' hay
    "LED=OFF"

    {

    digitalWrite(sirena,LOW);

        estado="OFF";

    }

    //Cuando reciba una línea en blanco, quiere decir que la petición HTTP ha acabado
    y el servidor Web está listo para enviar una respuesta

    if (c == '\n' &&currentLineIsBlank) {

    // Enviamos al cliente una respuesta HTTP

    client.println("HTTP/1.1 200 OK");

    client.println("Content-Type: text/html");

    client.println();

    //Página web en formato HTML

    client.println("<html>");

    client.println("<head>");

    client.println("</head>");

    client.println("<body>");

    client.println("<h1 align='center'>UNIVERSIDAD NACIONAL DE PIURA</h1><h3
    align='center'>Desarrollo de un sistema de supervision y control de una planta usando
    Comunicacion Ethernet y TecnologiaMovil</h3>");

    client.println("<h1 align='center'>ESTACION 1 : MODULO DE CONTROL DE
    SEGURIDAD</h1><h3 align='center'>Control de la sirena :</h3>");

```

```

//Creamos los botones. Para enviar parametros a través de HTML se utiliza el
metodo URL encode. Los parámetros se envían a través del símbolo '?'

client.println("<div style='text-align:center;'>");

client.println("<button                                onClick=location.href='./?LED=ON\'
style='margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border: 1px
solid #3F7CFF;width:65px;'>");

client.println("ON");

client.println("</button>");

client.println("<button                                onClick=location.href='./?LED=OFF\'
style='margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border: 1px
solid #3F7CFF;width:65px;'>");

client.println("OFF");

client.println("</button>");

client.println("<br /><br />");

client.println("<b>ESTADO DE LA SIRENA = ");

client.print(estado);

client.println("<META                                HTTP-EQUIV=Refresh
CONTENT=0.5;URL=http://192.168.1.10>"); //línea de código que actualiza todo el
contenido web del host

client.println("<br /><br /><br />");

client.println("<b>LECTURA DE LA DISTANCIA EN EL SENSOR ULTRASONICO:
");

client.print(distancia);

client.println("<br /><br /><br /><br />");

```

```

client.println("<b>LECTURA DEL SENSOR DE MOVIMIENTO : ");
client.print(movimiento);
client.println("</b><br />");
client.println("</b></body>");
client.println("</html>");
break;
    }
    if (c == '\n') {
currentLineIsBlank = true;
    }
    else if (c != '\r') {
currentLineIsBlank = false;
    }
    }
    }

    //Dar tiempo al navegador para recibir los datos
delay(1);

client.stop();// Cierra la conexión
}
}

```

## **CODIGO EN ARDUINO DEL MODULO DE IDENTIFICACION Y CONTROL DE ACCESO:**

```
#include <SPI.h> //Importamos librería comunicación SPI para el uso del puerto ICSP
#include <Ethernet.h> //Importamos librería Ethernet de la Shield W5100
#include <MFRC522.h> //Importamos librería para usar el módulo RFID
#include <Servo.h> //Importamos librería para trabajar con un servo motor
#include <Wire.h> //Librería que permite la comunicación con dispositivos SPI

Servo myservo;

int pos=0;

/*
Pines de conexión del SPI:      Arduino UNO

1 (NSS) SAD (SS)  10

2   SCK   13

3   MOSI  11

4   MISO  12

5   IRQ                      No conectar

6   GND   GND

7   RST   9

8   +3.3V (VCC) 3V3

*/

#define SAD 10 //Definimos la configuración de pines del RFID

#define RST 9

MFRC522 nfc(SAD, RST);
```

```

#define ledPinAbierto 5

#define ledPinCerrado 6

//Configuración de la red:

byte mac[] = { 0x00, 0x22, 0x48, 0x9F, 0xE2, 0x3D }; //Ponemos la dirección MAC de
la Ethernet Shield

IPAddress ip(192,168,1,20); //Asignamos la IP al Arduino

EthernetServer server(80); //Creamos un servidor Web con el puerto 80 que es el puerto
HTTP por defecto

String estado="OFF"; //Estado del servomotor, inicialmente: "OFF"

void setup()
{
myservo.attach(3);

  pinMode(ledPinAbierto,OUTPUT); //declaramos el modo del puerto para un LED
pinMode(ledPinCerrado,OUTPUT);

  SPI.begin();

  Serial.println("BUSCANDO MFRC522."); //comparamos el firmware del módulo
nfc.begin();

byte version = nfc.getFirmwareVersion();

  if (! version) {

    Serial.print("NO SE ENCONTRO MFRC522 ");

    while(1); halt

  }

  myservo.write(200);

  Serial.print("BUSCANDO CHIP MFRC522 ");

```



```

Serial.print("FIRMWARE VERSION. 0x");

Serial.print(version, HEX);

Serial.println(".");

Serial.begin(9600); //sincronizamos la velocidad del puerto serial: 9600 baudios para
el módulo Arduino UNO

//Inicializamos la comunicación Ethernet y el servidor

Ethernet.begin(mac, ip);

server.begin();

Serial.print("El servidor se inicializa con la IP: ");

Serial.println(Ethernet.localIP());

pinMode(orden,OUTPUT);

}

#define TARJETA 1 //declaramos la función para el arreglo lineal que conforma la
dirección MAC de los tags

#define LLAVE 1

//CLAVE DE LA TAJETA 1

byte Autorizado[TARJETA][4] = {{0x07, 0x0C, 0x75, 0x65 }};

// CLAVE DE LA TARJETA 2

byte Autorizado2[TARJETA 2][4] = {{0x44, 0x86, 0x58, 0x06 }}; //el número 4 que
vemos entre llaves, indica la cantidad de pares que necesitamos para identificar la
dirección MAC del tag

void imprimeClave(byte *serial);

boolean esIgual(byte *key, byte *serial);

boolean chekaKey(byte *serial);

```

```

void loop()

{

    byte status;

    byte data[MAX_LEN];

    byte serial[5];

    boolean Abierto = false;

    digitalWrite(ledPinAbierto, Abierto);

    digitalWrite(ledPinCerrado, !Abierto);

    status = nfc.requestTag(MF1_REQIDL, data);

    myservo.write(200);

    if (status == MI_OK) {

        status = nfc.antiCollision(data);

        memcpy(serial, data, 5);

        if(chekaKey(serial)){

            Serial.println("AUTORIZADO");

            imprimeClave(serial);

            Abierto = true;

            myservo.write(90);

        }

        Else

        {

            imprimeClave(serial);

            Serial.println("NO AUTORIZADO");

            Abierto = false;

```

```

    }

    nfc.haltTag();

    digitalWrite(ledPinAbierto, Abierto);

    digitalWrite(ledPinCerrado, !Abierto);

    delay(1500);

}

delay(500);

}

boolean esIgual(byte *key, byte *serial){

    for (int i = 0; i < 4; i++){

        if (key[i] != serial[i]){

            return false;

        }

    }

    return true;

}

boolean chekaKey(byte *serial)

{

    for(int i = 0; i<TARJETA; i++)

    {

        if(esIgual(serial, Autorizado[i]))

            return true;

    }

    for(int i = 0; i<LLAVE; i++)

```

```

    {
        if(esIgual(serial, Autorizado2[i]))
            return true;
    }

    return false;
}

void imprimeClave(byte *serial)
{
    Serial.print("CLAVE: ");
    for (int i = 0; i < 4; i++) {
        Serial.print(serial[i], HEX);
    }
    Serial.print(" ");
}

//Programación del Módulo Ethernet:

EthernetClient client = server.available(); //Creamos un cliente Web

//Cuando detecte un cliente a través de una petición HTTP
if (client)
{
    Serial.println("new client");

    boolean currentLineIsBlank = true; //Una petición HTTP acaba con una línea en blanco

    String cadena=""; //Creamos una cadena de caracteres vacía

    while (client.connected()) {
        if (client.available()) {
            char c = client.read(); //Leemos la petición HTTP carácter por carácter

```

```

Serial.write(c);//Visualizamos la petición HTTP por el Monitor Serial

cadena.concat(c);//Unimos el String 'cadena' con la petición HTTP (c). De esta
manera convertimos la petición HTTP a un String

//Ya que hemos convertido la petición HTTP a una cadena de caracteres, ahora
podremos buscar partes del texto.

int posicion=cadena.indexOf("SERVO="); //Guardamos la posición de la instancia
"SERVO=" a la variable 'posicion'

if(cadena.substring(posicion)=="SERVO=ON")//Si a la posición 'posicion' hay
"LED=ON"

{

    //digitalWrite(servo,HIGH);

    estado="ON";

}

if(cadena.substring(posicion)=="SERVO=OFF")//Si a la posición 'posicion' hay
"LED=OFF"

{

    digitalWrite(servo,LOW);

    estado="OFF";

}

//Cuando reciba una línea en blanco, quiere decir que la petición HTTP ha acabado y el
servidor Web está listo para enviar una respuesta

if (c == '\n' && currentLineIsBlank)

{

    // Enviamos al cliente una respuesta HTTP

```

```

client.println("HTTP/1.1 200 OK");

    client.println("Content-Type: text/html");

client.println();

//Página web en formato HTML

client.println("<html>");

    client.println("<head>");

client.println("</head>");

    client.println("<body>");

client.println("<h1 align='center'>UNIVERSIDAD NACIONAL DE PIURA</h1><h3
align='center'>Desarrollo de un sistema de supervision y control de una planta usando
Comunicacion Ethernet y Tecnologia Movil</h3>");


    client.println("<h1 align='center'>ESTACION 2 : MODULO DE ACCESO VIA
RFID</h1><h3 align='center'>ESTADO DEL SERVOMOTOR :</h3>");

    //Creamos los botones. Para enviar parametres a través de HTML se utiliza el
metodo URL encode. Los parámetros se envían a través del símbolo '?'

    client.println("<div style='text-align:center;'>");

client.println("<buttononClick=location.href='./?SERVO=ON'
style='margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border: 1px
solid #3F7CFF;width:65px;'>");

    client.println("ON");

    client.println("</button>");

    client.println("<buttononClick=location.href='./?SERVO=OFF'
style='margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border: 1px

```

```

solid #3F7CFF;width:65px;>");

    client.println("OFF");

    client.println("</button>");

client.println("<br /><br />");

    client.println("<b>LECTURA RFID = ");

    client.print(lectura);

    client.println("<META HTTP-EQUIV=Refresh

CONTENT=0.5;URL=http://192.168.1.20>"); //línea de código que actualiza todo el
contenido web del host

    client.println("<br /><br /><br />");

    client.println("<b>LECTURA DEL TAG ACEPTADO: ");

client.print(permitido);

    client.println("<br /><br /><br /><br />");

    client.println("<b>LECTURA DEL TAG DENEGADO : ");

client.print(no permitido);

    client.println("</b><br />");

client.println("</b></body>");

    client.println("</html>");

    break;

}

if (c == '\n') {

    currentLineIsBlank = true;

}

else if (c != '\r') {

```

```
        currentLineIsBlank = false;
    }
}
}

//Dar tiempo al navegador para recibir los datos
delay(1);

client.stop();// Cierra la conexión
}
}
```



## **CODIGO EN ARDUINO DEL MODULO DE CONTROL DE TEMPERATURA:**

```
#include <SPI.h> //Importamos librería comunicación SPI para el uso del puerto ICSP

#include <Ethernet.h> //Importamos librería Ethernet de la Shield W5100

#include "DHT.h" //cargamos la librería DHT

#define DHTPIN 2 //Seleccionamos el pin en el que se //conectará el sensor

#define DHTTYPE DHT11 //Se selecciona el DHT11 (hay //otros DHT)

DHT dht(DHTPIN, DHTTYPE); //Se inicia una variable que será usada por Arduino
para comunicarse con el sensorR

// Declaracion de variables

float tempC;

int tempPin = 1; // Definimos la entrada en pin A0

dht.begin(); //Se inicia el sensor

//Configuración de la red:

byte mac[] = { 0x00, 0x22, 0x48, 0x9F, 0xE2, 0x3D }; //Ponemos la dirección MAC
de la Ethernet Shield

IPAddress ip(192,168,1,30); //Asingamos la IP al Arduino

EthernetServer server(80); //Creamos un servidor Web con el puerto 80 que es el
puerto HTTP por defecto

String estado="OFF"; //Estado del la sirena inicialmente "OFF"

void setup()

{

{
```

```

// Lee el valor desde el sensor

tempC = analogRead(tempPin);

// Convierte el valor a temperatura

tempC = (5*tempC*100)/1024.0;

// Envía el dato al puerto serial

Serial.print(tempC);

Serial.print(" grados Celsius\n");

// Espera cinco segundos para repetir el loop

delay(1000);

}

{

float h = dht.readHumidity(); //Se lee la humedad

float t = dht.readTemperature(); //Se lee la temperatura

//Se imprimen las variables

Serial.println("Humedad: ");

Serial.println(h);

Serial.println("Temperatura: ");

Serial.println(t);

delay(1000); //Se espera 1 segundo para seguir leyendo //datos

}

Serial.begin(9600); //sincronizamos la velocidad del puerto serial: 9600 baudios

para el módulo Arduino UNO

//Inicializamos la comunicación Ethernet y el servidor

Ethernet.begin(mac, ip);

```

```

server.begin();

pinMode(cooler,OUTPUT);

void loop()
{
    //Programación del Módulo Ethernet:

    EthernetClient client = server.available(); //Creamos un cliente Web

    //Cuando detecte un cliente a través de una petición HTTP
    if (client)
    {
        boolean currentLineIsBlank = true; //Una petición HTTP acaba con una línea en
        blanco

        String cadena=""; //Creamos una cadena de caracteres vacía

        while (client.connected()) {
            if (client.available()) {
                char c = client.read();//Leemos la petición HTTP carácter por carácter

                Serial.write(c);//Visualizamos la petición HTTP por el Monitor Serial

                cadena.concat(c);//Unimos el String 'cadena' con la petición HTTP (c). De esta
                manera convertimos la petición HTTP a un String

                //Ya que hemos convertido la petición HTTP a una cadena de caracteres, ahora
                podremos buscar partes del texto.

                int posicion=cadena.indexOf("COOLER="); //Guardamos la posición de la
                instancia "COOLER=" a la variable 'posicion'

                if(cadena.substring(posicion)=="COOLER=ON");//Si a la posición 'posicion'
                hay "COOLER=ON"

```

```

{
digitalWrite(cooler,HIGH);

    estado="ON";
}

    if(cadena.substring(posicion)=="COOLER=OFF")//Si a la posición 'posicion'
hay "COOLER=OFF"

    {
digitalWrite(cooler,LOW);

        estado="OFF";

    }

    //Cuando reciba una línea en blanco, quiere decir que la petición HTTP ha
acabado y el servidor Web está listo para enviar una respuesta
if (c == '\n' && currentLineIsBlank) {
// Enviamos al cliente una respuesta HTTP
client.println("HTTP/1.1 200 OK");

    client.println("Content-Type: text/html");
client.println();

//Página web en formato HTML
client.println("<html>");

    client.println("<head>");

    client.println("</head>");

    client.println("<body>");

```

```
client.println("<h1 align='center'>UNIVERSIDAD NACIONAL DE  
PIURA</h1><h3 align='center'>Desarrollo de un sistema de supervision y control  
de una planta usando Comunicacion Ethernet y Tecnologia Movil</h3>");
```

```
client.println("<h1 align='center'>ESTACION 3 : MODULO DE  
CONTROL DE TEMPERATURA</h1><h3 align='center'>Control del cooler  
:</h3>");
```

//Creamos los botones. Para enviar parametros a través de HTML se utiliza el  
metodo URL encode. Los parámetros se envian a través del símbolo '?'

```
client.println("<div style='text-align:center;'>");
```

```
client.println("<button onlick=location.href='./?COOLER=ON'  
style='margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border:  
1px solid #3F7CFF;width:65px;'>");
```

```
client.println("ON");
```

```
client.println("</button>");
```

```
client.println("<button onlick=location.href='./?COOLER=OFF'  
style='margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border:  
1px solid #3F7CFF;width:65px;'>");
```

```
client.println("OFF");
```

```
client.println("</button>");
```

```
client.println("<br /><br />");
```

```
client.println("<b>ESTADO DEL COOLER = ");
```

```
client.print(cooler); client.println("<META HTTP-EQUIV=Refresh  
CONTENT=0.5;URL=http://192.168.1.30>"); //línea de código que actualiza todo  
el contenido web del host
```

```

        client.println("<br /><br /><br />");
        client.println("<b>LECTURA DE HUMEDAD: ");
        client.print(humedad);
        client.println("<br /><br /><br /><br />");
        client.println("<b>LECTURA DE TEMPERATURA : ");
        client.print(temperatura);

        client.println("</b><br />");
        client.println("</b></body>");
        client.println("</html>");
        break;
    }

    if (c == '\n') {
        currentLineIsBlank = true;
    }

    else if (c != '\r') {
        currentLineIsBlank = false;
    }
}

}

//Dar tiempo al navegador para recibir los datos
delay(1);

client.stop();// Cierra la conexión
}
}

```